1993

Traffic profiles for audio and data collaborative work systems

W. T. Musangeya
University of Wollongong

Traffic Profiles
For Audio and Data
Collaborative Work Systems

A thesis submitted in partial fulfilment of the

requirements for the award of the degree of

Honours Master of Engineering

(Telecommunications Engineering)

from

UNIVERSITY OF WOLLONGONG

by

W.T. Musangeya

BSc(ENG)Hons.(UZ), AMIEE(UK)

elec952, 1993

Abstract

A predictive model of the behavior of multimedia communications systems on digital broadband networks is needed because, it is through knowledge of the traffic characteristics of the arrival process, that proper network resource dimensioning can be performed to avoid congestion and maintain the network quality of service. This thesis analyses the traffic from multimedia systems that are expected to become common in the next decade. The traffic is used to develop models to characterise the statistical properties of such multimedia communications systems.

The analysis was carried out on a real-time computer-supported collaborative work system, running on ethernet, to which an audio channel had been added. The audio channel enabled hands-free audio conversations during conferencing sessions. Data was collected as observations of the transmission time and the volume of information transferred. The observations were for the networking interface between the application layer and the operating system.

The model was described in terms of the variability of the packet arrival process and the degree of correlation between packet arrivals. The squared coefficient of variation was used to measure the variability of the arrival process. The variables investigated included interarrival time intervals and the transmission intensity of the arrival process. The model was presented in two parts: one describing the

individual packets' distribution and the second part looking at traffic bursts.

The model of the traffic bursts, for both the audio and data arrival process, was found to approximate a compound poisson process, with the burst interarrival time distributions indicating negative exponential curves. In the transmitted audio traffic, the model showed burst distributions consisting of sums of exponentially distributed burst interarrival time and transmission intensity curves. The distributions, for the packet model, were scattered over the time intervals from 5ms to the threshold value, showing less tendency towards any common distributions. This was observed for both audio and data packet interarrival times and packet transmission intensity.

The significance of this work is in relation to the modelling of multimedia networks. It has been shown[Habib 92] that the different types of data sources transported on multimedia networks can each be modelled as individual Markov chains. The model presented in this thesis describes the statistical properties of the variables that would require specification in the Markov chain model. The distribution curves for the transmission intensity provide for the peak and average rates, whilst the burstiness parameter describes the time-scale rate of variation.

Acknowledgements

I want to express my gratitude to my supervisor, Dr. HWP Beadle, without whose support and assistance, this study would not have been feasible. I am grateful for his patience in answering the many questions I had throughout this study, and explaining most of the concepts I didn't understand.

I would also like to express my appreciation to the Australian International Development Assistance Bureau (AIDAB) for their financial support, and to the University of Wollongong, for providing an optimal study environment.

I also thank the wonderful people at the International Council for Overseas Students who made my stay in Australia an enjoyable experience; and to the many friends I made, for their moral support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The design of computer networks involves the design of the transmission circuits between the hosts as well as the design of the switching connecting any two or more transmission lines within the network. Switching routes data arriving on an incoming circuit to the appropriate outgoing line for forwarding to it's destination.

The design of transmission circuits can be for point-to-point communication or broadcast communication channels. In the former, a circuit is used to connect two switches so that only indirect communication can be used between hosts who are not directly connected by a circuit. Broadcast, on the other hand, requires a single circuit that is common to all the hosts. In this case the transmitted messages are received by all connected computers and each host only copies the messages if the messages are directed to that host's address.

Other issues involved in computer network design are

- error detection and error correcting mechanism
- preservation of the order that the messages are sent.
- method for establishing connection with the desired host and terminating the call when done.
- method of data transfer i.e. unidirectional or full duplex operation.
- control of the rate of receiving or sending messages i.e. flow control.
- limits on the length of messages that can be handled i.e. assembling and disassembling of messages.
- routing decisions where more than one route exists between the hosts requiring data transfer.
- channel and bandwidth allocation.
- contention problems in shared channel systems.

1.1 Statistical analysis

In all these design issues, the characterisation of the carried computer signals takes an important part. For example, to properly allocate bandwidth, it is necessary to know the probability of a frame being generated in an interval, say, δt. This probability is calculated as $\lambda \delta t$ where λ is the arrival rate of new frames.

It is therefore essential to understand the statistical properties of the signals so that predictions can be made on the variation of the data.

The predictions possible with the statistical information make it easy to allocate network resources in anticipation of demand. This is very desirable especially with the imminent broadband intergrated services digital networks (ISDN) and the asynchronous transfer mode (ATM) switching technology which are designed to offer bandwidth on demand. A brief description of broadband ISDN is given in section C.1.

1.1.1 Motivation

The statistical analysis carried out in this project is an attempt to change the sample numerical traffic data into meaningful facts that can aid in network design decision making. From this analysis, generalizations can be made with regard to the traffic profiles of similar computer applications.

The use of graphs and tables to present the collected data and the deduced traffic profiles, gives a clearer, easy to understand picture of the traffic characteristics. The basic statistical element in the investigation will consist of ratio data in the form of transmission time and the volume of the information transferred. From this data, the time interval data is derived.

The ratio data is obtained from several executions of shared workspace applications. A shared workspace is defined[Guan 88] as a collection of objects belonging to some work group and the software tools that are required for their manipulation. Limitations of time and the large varieties of available shared workspace programs, make it impossible to investigate the complete population. Thus, only a sample is used, consisting of observations of one particular shared workspace application. Shared workspace applications are synonymous with Computer Supported Collaborative Work(CSCW) systems. CSCW systems allow joint use of computer based material, providing multi-point computer conferencing. They achieve this by presenting a common view of the work surface to which simultaneous access is given to the connected participants. Several CSCW products have been developed; a few of them are described in section 3.1. Section 2.5.2 discusses CSCW in more detail.

CSCW sessions can be considered "Multimedia" since they involve, in addition to the CSCW applications, audio conferences and sometimes video conferences. Given these different types of data, all carried on one multimedia network, quality of service(QOS) becomes vital in preserving the communication content. It has been shown[Sriram 86] that packetized voice communication is sensitive to jitter and that it's quality decreases with an increase in the system response times. This is especially true when the network is congested. Data communication, on the other hand, requires stringent error characteristics. These are conflicting requirements and hence the need for investigation into the statistical nature of the traffic generated in systems where audio and data are combined.

So far research on shared workspaces has concentrated on implementation, resulting in a wide range of cscw products being developed. With all these product developments, it is time that attention focussed on the statistical nature of the network traffic generated by these new applications. This research is necessary in order to design networks that are well qualified to carry multimedia and CSCW traffic. The resulting traffic models would then aid networked multimedia system design to better dimension the network. This would provide information on how well these systems work on the existing networks or information on which proposed new networks are better suited to these systems.

1.2 Objectives

The objective of this research is to collect and analyse the statistics of a collaborative work system where several users receive audio and visual information simultaneously. The objectives are to design the system, measure the performance and characterize the performance to achieve an overall goal of understanding the traffic characteristics of the service. The network statistics of the traffic that ensue are then analysed to develop a workable traffic model.

The conferencing system is developed on the Unix operating system using the X Window to run on the Sun SparcStation. The audio channel in this system is provided through the audio input and output facilities offered by the Sun SparcStations. It is designed to allow communication between two or more users. This system offers flexibility and the ease of use that comes with voice conversations, e.g. psychological cues in the tone of voice, length of pauses, etc, in addition to the ability to share visual material through the computer terminal. CSCW data is carried on Internet's connection oriented Transmission Control Protocol (TCP) stack, whilst the audio uses the connectionless User Datagram Protocol (UDP) stack. The collaborative work system does not use the broadcast and multicast protocols, relying only on polling and sequential delivery of data.

The research uses ethernet because of it's availability on a larger scale. It is one of the most widely used networks and it presents the worst case in network performance due to the influence of the network's backoff algorithms which introduce undesirable delay jitter on voice packets. Ethernet does not offer any priotization of voice packets over data packets.

1.2.1 CSMA/CD Networks

Ethernet is a carrier sense multiaccess bus network (CSMA/CD) with collision detection. It works by having the source hosts detecting the carrier on the bus. Absence of carrier signal on the bus is interpreted as a go - ahead for transmission. When the host senses the absence of carrier it acquires the bus by transmitting it's information. All other hosts then wait until the transmission is completed. If two or more hosts transmit simultaneously i.e. a collision occurs, both stop transmission and wait a random time before attempting retransmission. Collisions are detected when the sending hosts read back their information from the bus to check for correct transmission. In case of collisions, instead of receiving it's own information a host receives a mix of it's information and the other host's information. The detecting host then jams the bus so that all intending source hosts have to backoff for a random time that may be determined by some function e.g. the binary exponential backoff algorithms. A typical slot time of 38us round trip bit propagation time[Stallings 89] is the time it takes for the first bit of the second host's transmission to travel to the first host.

1.3 About the report ...

The report begins with brief background information on past research into traffic characterisation in Chapter 2. This chapter also introduces the topics of groupware and multimedia networks.

Chapter 3 presents an overview of the CSCW systems that have been developed so far, and presents a detailed description of the developed cscw system's technical architecture.

Chapter 4 gives the procedures used to gather the traffic statistics, the environment in which the experiments were carried out and the analysis of the results that were obtained.

Next is discussed the modelling methods employed, leading to a summary and the presentation of the developed traffic model in Chapter 5.

Chapter 6 gives examples of situations to which the model can be applied.

Chapter 7 concludes the research followed by the bibliography list and a set of graphs in the appendix.

Chapter 2

Background

The statistical analysis of computer-generated traffic has been carried out over the years by several researchers. Though none of the work investigated the traffic generated from shared workspaces, most of the research has been on the performance of available networks (ethernet, token ring[Yang 92], etc) in transporting a combination of different traffic types like audio and data traffic[Nutt 82] or voice and video traffic[Habib 92]. There has also been some research on modelling data as groups of packets[Jain 86].

2.1 Broadband Networks

Traffic profiles for voice and video traffic sources have been investigated[Habib 92] for use on broadband networks. In their investigations, Habib and Saadawi expressed how the variability of the variance lead to queueing delays, contributing to congestion. The aim was to find analytic models to characterise correlation and burstiness of multimedia traffic. They described the voice process as a bursty Markov process, quoting the squared coefficient of variation(SCOV) figure of 18.1 from [Sriram 86]. Bursty traffic was defined as *one that exhibits a high degree of variability compared to that of the poisson process.* The packet arrivals during a talk spurt were modelled as a Bernoulli distribution and the duration of each talk or silence state modelled as a geometric distribution. For multiple voice sources, a markov chain of M states was used with the state as the number of voice sources in the talk state. To represent sources with different traffic characteristics (-*multimedia traffic*), independant and identically distributed Markov chains for each source were suggested. The SCOV, as defined in [Sriram 86], was applied to the continuous time model of the arrival process and the index of dispersion for counts (IDC) used for dicrete-time model. The IDC is defined as

$$I(t) = \frac{\mathrm{var}(N_t)}{E(N_t)}$$

with N_t as the number of arrivals in an interval length. It is the variance of the number of arrivals in an interval t normalized by the average number of arrivals in that interval length.

2.2 Ethernet

Nutt and Bayer [Nutt 82] covered the performance of ethernet on a combined voice and data load. A simulation model of the ethernet network was used. Their experimentation was concerned with adapting ethernet to carrying combined voice and data effectively and efficiently. They tested their model using overload conditions specified by Metcalfe and Boggs[Metcalfe 76] and the packet size and interarrival time observed by Shoch and Hupp[Shoch 80].

They formulated two types of networks; one that distinguished between voice and data traffic and the other which didn't, and adopted different backoff algorithms for each type of network. Acknowledging that data can tolerate delays during congestion periods, and that voice packets have real time limits, a random algorithm was applied to voice and a binary exponential algorithm to data packets. The random algorithm dynamically determined the backoff time using a uniform distribution function which was sampled by some predetermined value. The binary exponential backoff algorithm has backoff times growing exponentially and allows for congestion conditions, achieving recovery of the network through degradation of the network performance. Another suggestion given was the use of the binary exponential algorithm for both voice and data packets but with twice the value of the exponential distribution mean for data packets. The experiments covered in the paper suggested that both the above algorithms failed under heavy traffic conditions during voice applications, resulting in delays greater than their threshold defined as 5ms. Their tests involved simulated loads instead of actual data, with approximated interarrival time and packet size distributions.

2.3 Traffic into a Multiplexor

There has also been research covering the analysis of the performance of packetized voice and data traffic on a statistical multiplexor[Heffes 86]. The number of arriving voice packets is modelled as a geometrical distribution with exponentially distributed talk and silence states, in an interval of length 16ms. From this model and using the Laplace-Stieltjes transform (LST) [Heffes 86], it is shown that the SCOV of the interarrival time is given by

$$SCOV = \frac{var(X)}{E^2(X)}$$

for both a single voice source and a superposition of an arbitrary n voice sources. The combined arrival process of packetized voice and data streams is modelled as a Markov Modulated Poisson process. This is a stochastic process where the arrival rate λ_j of the process is equal to the state of a continuous time two-state Markov chain. i.e. the arrival rate of the process is equal to the Poisson arrival rate of the current state j. The parameters specified in the model are the mean arrival rate (noting that λ^{-1} is the *mean time between arrivals*), the variance to mean ratio of the arrivals in an interval, the same ratio long-term, and the skewness parameter given by the third moment of the number of arrivals in an interval. Simulations of the model provided results similar to those of a deterministic process, as described in [Sriram 86]. This was using fixed voice packet lengths of 64 bytes, geometrically distributed data packet lengths with mean 50 bytes, and data packets arriving as a Poisson process. The paper also observes that the correlation structure of the superposition structure is defined well by the variance - time graph.

2.4 Voice systems

Models to describe the statistical properties of packet voice systems were investigated by Daigle and Langford [Daigle 86]. Their paper models the voice packet generation process as a Poisson process with the number of active voice sources represented as a continuous-time Markov chain. It also investigates fixed packet generation rate with first, a semi-Markov process and then, with a uniform arrival model. The analysis was strongly oriented towards developing a queueing system model rather than modelling the packet generation and subsequent arrival process parameters.

2.5 CSCW systems

CSCW systems are arguably a new application that is rapidly expanding in recent years. More and more CSCW systems are being integrated with audio and video information for transmission on a single *multi-media* network. It is therefore worthwhile to look at the behaviour of multimedia networks to enhance our understanding of the traffic profiles from CSCW systems. Multimedia networks provide an integrated communication media for data sources. They range from the integrated transportation of audio and data on a data network to the transportation of traffic from disparate data sources on a single network. The following literature gives a brief discussion on traffic characteristics of multimedia networks, and introduces CSCW systems as a branch of multimedia networks. This literature should give an incite into what to expect for the traffic analysis in this research.

2.5.1 Multimedia Networks

The diversity of the data sources on multimedia networks may include variations in the speed of data, deviations in the length of the data and variations in the rate of arrival of the data packets[Schwartz 77]. This is a different scenario from bulk data transfer where transfer occurs at an average bit rate and normally consists of data of the same type. The main information sources in this scenario are voice, video, graphics, and high quality audio. The goal is to achieve integration and synchronization without performance degradation. This sets limits on the types of data sources that can be transported as well as on the number of data sources that can be combined at any one time. Traffic profiles have to be described in order to model such points of integration.

Simple examples of systems that would require integrated communication media and hence are multimedia are adding video to electronic mail or moving video conferencing into a window on a computer screen.

Multimedia information flows can be grouped into three types.

- User to document information flows as in e-mail
- User to computer as in information systems, accessing databases through graphical user interfaces.
- User to user information flow as in CSCW systems accessed through conferencing or training sessions.

Multimedia network characteristics

As multimedia systems carry voice and video, which are essentially continuous communication media, they therefore require continous data transfers over long periods of time.

In order to keep each presentation device's fixed data ratio, multimedia networks may require event-driven or on-going synchronization relationships between the real time data channels. An example of where this would be necessary is the transportation of a moving picture signal, where the separately stored audio signal is correlated with the video signal and the trasmission requires synchronization to maintain intelligibility.

Because of the wide variety of traffic types transported in multimedia networks, a particular QOS is required. A dynamic supply of network resources like network bandwidth, processor time or disc bandwidth should be guaranteed. Provision should be made for control of this QOS so that it is possible, for instance, to sacrifice the QOS or to reject a request for service in the face of insufficient resources. These are called call admission requirements.

2.5.2 Groupware

Computer-based group activities are found in the area of multimedia applications that consist of user to user information flows. These are presented as real time computer conferencing systems with a variety of tools added in; tools like integrated voice and data communication, joint-editing facilities and collaboration tools. Further information on computer conferencing system is given in appendix B. The objective is to provide interaction similar to that which occurs in face to face drawing processes. These computer based group activities are referred to as groupware, CSCW systems or shared workspaces. They provide integrated support across group activities regardless of the users' geographical locations, coordinating the dynamic sharing of screen tools in a real time nature. The data sources that groupware may combine are computer data, text, and remote image service feeds. It is this ability to support collaborative projects across the users' networks that may make it the pillar of activities in Publishing Houses, Military Agencies and R & D Laboratories[McQuillan 92].

Group interaction requires continuous media transmission for applications requiring simultaneous display of information on several displays. This is called group communication. In other cases, a single operation applied to a number of displays may be required and this is an example of group invocation.

The main applications of groupware are in conferencing [McQuillan 92] and training. Groupware requires real time two way transmission with high bandwidth and a guaranteed QOS – i.e. low delay with low variance and low data loss. The successful acceptance of groupware is based on it's ability to combine existing individual work practices with the collaborative work mode.

In designing groupware, it is necessary to identify the members of a work group that have a joint need to communicate. A deep understanding of the human characteristics is required, as is the organisational aspects like the structure of an organisation. The involvement of the user in the design process is important. Group dynamic aspects like decision making or the collaboration process must be considered [Johansen 88]. Collaboration can be by group invocation or group communication.

Video conferencing and electronic mail are examples of the communication mechanisms that support group work. Like groupware, they offer shared workspace facilities, shared information facilities as in multi-user databases, and facilities that augment specific group work processes like co-authoring of documents or idea generation[Hopper 91].

Groupware, like everything else, has to undergo usability testing for wider acceptance. It demands connectivity and availability at all times. These are the factors that increase the efficiency of searching and exchanging information. An example is given[Brand 88] of a user who multicasted an e-mail asking for information on a topic and received several responses from all over the world the next day. Example uses of such systems would be in the circulation of memos and reports, and for group revision or group review of some subject matter.

On-line editing is another widely accepted form of groupware. It finds applications in message systems, procedure processing systems, screen sharing systems and calendar systems.

2.6 Summary

Looking at the research that has been carried out in traffic characterisation, it is clear that most models of packetized voice signals have been presented as Markov chains. This has been due to the ease with which the Markov Chains are applied to queueing network analysis. The missing information required now is how the Markov chain is defined for particular applications. This may be in the form of statistical properties for interarrival times, data rates and how bursty the arrival process is.

The application whose traffic characteristics are to be investigated has been introduced as a CSCW system with added voice communication. The next chapter discusses this system in detail and gives a brief description of other CSCW systems that have been developed.

Chapter 3

System under test

There have been several CSCW systems developed. The following sections describe some of the developed systems and gives the criteria used in choosing a CSCW system for traffic analysis.

3.1 Example systems

As mentioned in section 2.5.2, CSCW systems present a common view of the work surface allowing simultaneous access by the participating users. Several different shared views can be used. Below are a few examples.

3.1.1 Shared screen systems

Timbuktu remote is an example of a shared screen system [Farallon 91] which is also similar to the shared window system[Lauwers 90]. Multiuser editors like

GroupSketch[Greenberg 92] and shared text editors[Neuwirth 90] also use a shared screen as a common view.

3.1.2 Video drawing media

An example of a shared video drawing media is the TWS (TeamWorkStation), developed by Ishii and Miyake [Ishii 90], [Ishii 91]. The TWS integrates the computers and the desktop workspaces. It also provides distributed users with an open shared workspace. TWS is designed in such a way that the individual workspace images are overlaid. Each individual continues to use their application programs. At the time of writing their paper, TWS had not been tested with a larger variety of tasks. According to [Ishii 91], TWS is not aimed at replacing groupware. It supports a broader range of dynamic collaboration activities which are not supported by existing groupware. Their recent work[Ishii 93], *ClearBoard*, is a shared drawing medium that supports gaze awareness in remote collaboration.

3.1.3 Talk and Write

Talk is a visual communication program which copies text lines from one terminal to another user. It is run on Unix computers. Once connection is established, two users may type simultaneously with displays on separate windows.

talk uses a server *talkd* to listen at the udp port and a tcp connection is made for the conversation.

Write is a program similar to talk except that it uses the same screen for both users and does not cater for simultaneously sent messages. The users have to develop a system of letting the other know they are waiting for a response e.g. the use of *over* or *over and out.*

3.2 Integrated Voice And Data

Integration of voice and data is usually achieved through voice digitization. For better transmission quality and efficiency, the voice digitization at the source is then followed by the use of a digital transportation to the customers' premises.

An example product designed to handle both voice and data traffic is Netrunner[Salamone 92 developed by Micom. It is a system that uses data and speech compression to boost throughput. It also handles voice and facsimile traffic in addition to linking Local Area Networks.

Voiceview[Radish 92], is an application developed by Radish Communication System Inc. which can transmit data during a voice call on analog lines. One has to make a voice call and then downloads data by the pressing of a button.

3.3 Chosen system

In choosing a collaborative work system for traffic analysis, it was desired that it be representative of most other CSCW systems as well as being versatile and simple to use. Thus the following system(section 3.4) was chosen because it contains the basic building blocks of many other advanced system, and also because of it's suitability for an educational environment.

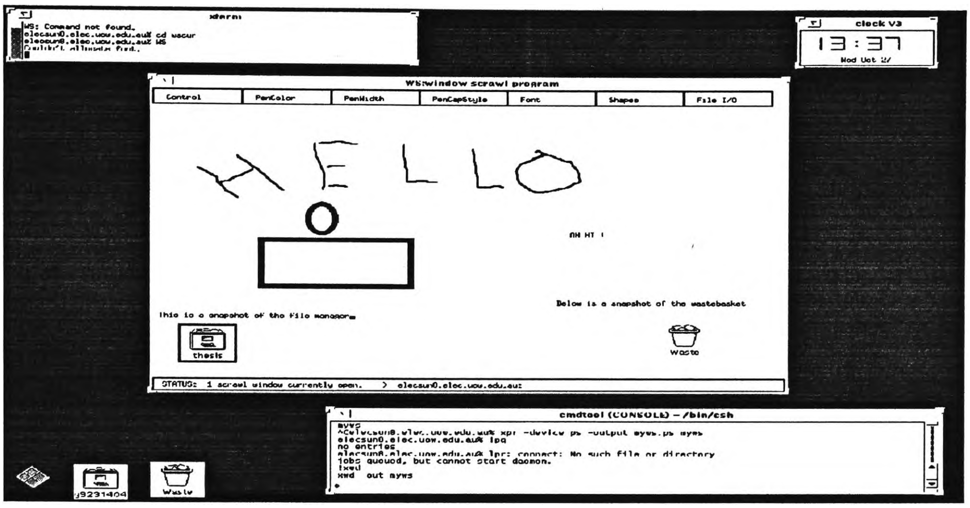

Figure 3.1: WS common window

3.4 WS

The developed collaborative work system, which will now be referred to as *ws*, is a real-time computer conferencing system, adapted from the window scrawl program *wscrawl* written by Brian Wilson[Wilson 92] of Hewlett-Packard. *Ws* was developed and tested within a university local area network environment on the SUN SparcStations. Figure 3.1 shows a workstation screen during a *ws* session. The *ws* window is the large one containing the hand drawn *HELLO.* The content of the *ws* window is shared with all the users in a session. All the other windows on the screen are private. The system is designed for use on the Unix operating systems connected to the internet domain. Figure 3.2 shows how *ws* uses the network. *ws* uses the sockets and the TCP/IP protocol stack to provide a reliable two way virtual circuit connection between several conference members. The required windows were created using the X Window system. The *time to live* address field was left set to the default of 60 seconds to allow all data to be sent regardless of delays of up to 1 minute. If a packet is delayed beyond this value it is discarded. This is desirable in order to stop transmission of such delayed packets whose reconstruction at the destination result in unintelligible conversation.

Ws is an X-Window based application that creates a collaborative computing environment by allowing controlled shared access to a few chosen objects. The program provides a common white board (see figure 3.1) in the form of a computer window, allowing participants to display images, scribble and use gestural expressions in referring to material displayed on the window. It thus provides facilities that would be required in any face-to-face meeting.

3.4.1 Additions to ws

Realizing the limitations of *ws* for collaborative work and tutorial presentation, the following features were added,

- the *snapshot* function; *Snapshot* allows the user to grab any window image on the screen and paste the image onto the common viewing area. It is implemented on the same principles as the window dump utility, *xwd*, but uses routines like *read-image-on-disk* which are already available on *ws* to process the images.

- the *slide show* function; *slide show* provides the means to store several images as files in a directory. The slides can then be displayed consecutively on the common viewing area for use in presentations. It is implemented on similar base to snapshot. It produces a menu that allows the user to make choices on the activities that can be done, activities like *store slide, show slide, next slide*, etc. When the slide show is running, only the user who initiated it has control. The rest of the participants can view the slide and use the audio channel for discussion. To allow the rest of the users to use gestural expressions on the displayed slide, the user *quits* the slide show. This leaves the current slide displayed and enables all the users to use the *ws* tools whilst reviewing the displayed slide.

- The size of the window was reduced to allow full display on the smaller Macintosh screen.

3.4.2 Starting a new session

The parameters required to start a collaborative work session are a list of the participants to be included in the collaboration and optional values to override the default settings of the creation of windows. These are specified at the command line on running the program. Here a session is defined as the participants engaged in the collaboration. Additional participants can only be added to an existing session by those already in session and this can be done at any point in the collaboration and by any connected participant.

For every display that is to be involved in the session, the *ws* program opens three windows on the screen; one for window control, one for displaying status information and a shared white board for displaying user activity. After initializing the windows, *ws* uses the *select* system call to monitor the input of all the displays' session windows for either, the occurrence of an event or timeout.

The events, monitored by the *select* function, can be input from the keyboard, the mouse, the control window (-for resizing), the server, the socket, etc. When an event occurs, *ws* scans the users' displays for the event, and then performs the appropriate action. If the event involved typing, pointing or dumping of an image on the screen, *ws* replicates this action on every window opened by the program.

Functions for opening text files, reading or saving bitmaps and images are provided. One can also choose the pen colour or width and select style and font of the typed characters. A rubber pointer allows gestural expression through a unique label (i.e. the display name) that is visible on all displays, at the point of reference, making explanations clearer. In addition, *ws* provides basic functions for sketching, typing, erasing, drawing shapes and clearing windows.

3.4.3 Joining and leaving a session

Ws is executed at only one of the workstations involved in the collaboration and provides facilities for adding new participants to an already established session. One can also withdraw from the session without affecting the continuity of the CSCW session.

3.4.4 Contention Problems

Unlike some other systems[Guan 88], no token is provided to control the users' inputs to the common window. The only restraint available is the display of a *please wait* message displayed on all windows when the computer is busy. This occurs, for instance, when processing a request or updating the participants windows at times when large data transfers are necessary and are taking a long time. All participants have equal status during a session and hence can leave the session at any time without the need to elect a new *leader*[Guan 88].

The problem of more than one participant accessing the same tool at the same time is avoided by the fact that *ws* uses polling (and the *select* function) to process any event occurrence.

3.4.5 Session status and information

Information about the number of current displays open and their identities are displayed on the status window. This information is updated every time a user joins or leaves a particular session.

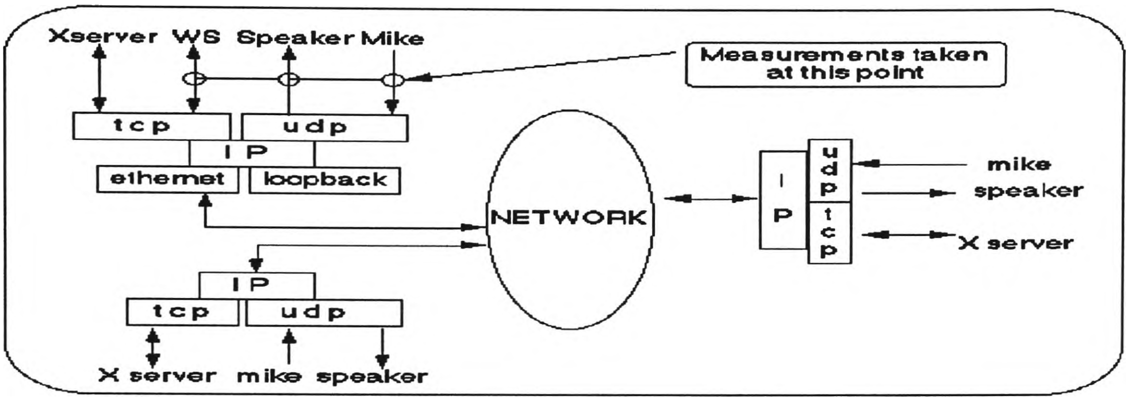

Figure 3.2: WS conferencing network

3.4.6 Limitations

Figure 3.2 shows the implementation of *ws* with the process and data centralized at the session creator's machine. Inputs typed by any user are forwarded to the session creator program and the output, generated from the session creator machine, is sent to all displays in the session. This requires larger bandwidths and poses security risks as users are allowed access to the session creator's directory.

Another limitation is in the design of *ws*, where the users cannot use system calls or system objects except those that have been incorporated into the application itself, thus only tools like file input/output or the slide show which are designed within *ws*, can be used.

Keeping the window display consistent is another source of problems. As already mentioned, *ws* displays a *please wait* flag on all displays whilst updating the users' displays. This results in larger delays or session response times. This problem brings out the superiority of replicated implementation over the centralized implementation. With replicated implementation[Guan 88], only the input from a user is sent to the session creator's machine. This machine then sends a message to all participants, so that each station regenerates the same input and has it's local program producing the output data locally. This, in addition to shorter response times, would require less bandwidth between the users' machines.

3.5 Mike & speaker

The audio component of the collaborative work system is provided by two programs, *mike* and *speaker* [Walker 91]. *Mike* and *speaker* are designed to work on the Sun SparcStations providing voice communication using the terminals' audio facilities to packetize the voice signal.

Mike does the recording and transmision of the voice signal. It supports both real time audio and pre-recorded sound samples, and uses the *record*() function. This function reads audio from the device file /dev/audio so that *mike* holds this device when it transmits data. Normally sound is transmitted at the standard rate of 8000, 8-bit samples per second, to give the 64kb/s rate.

Voice coding techniques, like silent interval detection or TASI (time assigned speech interpolation) can be used for bandwidth reduction to values as low as 16kb/s.

Mike, however, can be configured to transmit compressed audio at 4000, 8-bit samples per second reducing bandwidth to $\approx$ 32kb/s, a value suitable for transmission on 56kbit/sec links. This compression is achieved by transmitting only the even numbered sample points i.e. halving the number of samples. Speaker then does a linear interpolation between the received compressed audio points to reconstruct the full signal. The resulting lower quality audio is more intelligible than the effect of lost words and erratic pauses that would occur if the uncompressed signal was transmitted on lower bandwidth lines.

Speaker is the receive and playback component of the communication system responsible for the reconstruction of the sound samples. A workstation will not be able to receive audio communication if *speaker* is not running on that terminal. *Speaker* creates a socket, binds the socket to a port and then listens for connections to the socket to be made by the *mike* program. It then uses the soundtool facilities on the Sun Sparcstation, which write to the device file /dev/audio on receiving sound samples. It releases the device after a 20 second timeout if no samples are received. A gain control tool, *x_gaintool*() can be run in conjunction with *speaker* to permit interactive setting of the audio playback level. It can also be used to switch the received audio from the computer speaker to the headphone jack.

By setting the recording and squelch levels to allow the sending of data only when someone is speaking, it is possible to emulate a collection of speakerphones sharing a single connection in a conference. The results are intelligible only if one person speaks at a time; because *speaker* interleaves the audio received from multiple sources packet by packet. A mixer program would be required to remedy this.

Hangover detection and silence suppression were used wehen the *mike* and *speaker* data was collected. The *mike* program averages the signal on a collected voice sample, and compares the average against a threshold. When the sample average is below threshold, it is compared with the previous samples to determine if it represents the end of a *talk period.* In the case that the sample represents the end of a speech burst, the sample is transmitted. If a second sample is received next and found to have energy lower than the threshold, this second sample is taken as noise occurring during a *silence period* and is discarded. Hangover was implemented to prevent clipping of the voice signal at the end of sentences.

Both *Mike* and *speaker* use the datagram sockets in the internet UDP/IP domain, which does not give any flow control or acknowledgments. As voice communication is real time, retransmitting a lost packet (if at all possible) will only degrade the quality of the reconstructed sound. Lack of acknowledgments, on the other hand, are a disadvantage in that no warnings are given in the case of breakdown in the communication line, or the fact that *speaker* might not be running on the remote displays. No flow control is possible should the need arise to control the rate at which packets are dispatched during the session. The program releases the audio device if it does not receive packets within a minute. This allows some other applications to use the audio tool during the periods of inactivity. *Speaker* uses the *usleep* function to wait for the received packets to play before grabbing the next set of packets.

3.6 Summary

Detailed descriptions of the CSCW and audio systems designed have been given. Looking at the structures of the CSCW systems described, it is clear that ws provides the basic building blocks covering most aspects of CSCW systems, the exception being the video drawing systems. In the next chapter, the experiments carried out to collect the traffic data generated by the *ws* and the audio communication programs are reported.

Chapter 4

Numerical Data Collection

The measurement of performance of voice and data integration systems is centred on the preservation of good quality voice conversation. In this research quality was assessed subjectively by determining the degree of discomfort that the participants felt as regards clarity of reconstructed speech, the system delay, and how easily they felt comfortable with the application system.

One measurement variable is the burstiness of the arrival process which gives an indication of how variable the arrival process is. It is characterized by the burst length and the burst distribution in time. From this variable, the probability and intensity of an arrival can be derived.

The interarrival times between consecutive arrivals are critical in the reconstruction of the transmitted sound. This is because if arrivals suffer uneven, long delays they may result in lost packets which transform to speech gaps on reconstruction, rendering the conversation unintelligible. The mean and variance of the interarrival time intervals are used to characterise the performance of the network.

The delay experienced by a packet through the network can also be used to determine the appropriate value to be used in the *time to live* field in the internet datagrams, for example, so that the transmission of excessively delayed packets is stopped. Delays can be due to packet generation time, the queueing delay before departure, transmission time through the network and the reception time at the intended destination. The presence of gateways that introduce uneven delays in networks as well as network characteristics, like random backoffs after collisions in CSMA/CD networks, all add to the absolute delay value. Thus the chances of non-uniform delay or jitter being experienced from packet to packet are of high probability. The mean and variance of this type of delay are required to be low, to prevent disruption of voice communication. Data packets, in contrast to voice, are more sensitive to error control and recovery techniques than to delay. Because of these differences, it is desirable to give priority of transmission to the digital data resulting from voice digitization in relation to the rest of the digital data. This would mean that hosts transmitting voice information get more chances of transmitting than the rest of the machines. Methods to achieve this on Ethernet, are discussed in [Nutt 82] .

The probability of losing a voice packet due to network delay is measured by setting a threshold delay value above which, the resulting reconstructed sound is of an unacceptable quality. In [Malek 88], the value of this threshold delay, with reference to an absolute delay, was given as 150ms for a local area network. The higher the probability of losing a packet, the less desirable it is to use the voice application programs. The probability is more a function of the network conditions than of the application programs.

This chapter presents the traffic profiles and gives an analysis of the results obtained from the execution of the collaborative work application programs described in the previous chapter.

4.1 Assumptions

The following briefly introduces the traffic model assumed for the collaborative work application and describes the variables that arise from the model.

4.1.1 The voice arrival process

As discussed in the previous section, the traffic characteristics of voice packets differ from those of data packets. Voice communication involves real time transportation of data traffic and consists of periods of talking and periods of silence. In most systems, packets are generated in the talking state only.

For the voice interarrival time intervals, our analysis assumes exponential distribution with mean values λ and μ for the arrival process during the talk and silent states respectively. This means that the arrival process is a poisson process, i.e. has unpredictable talk states. Similar assumptions were done in [Yang 92] and [Habib 92]. It would be difficult to determine these mean values directly because the orders of magnitude of the time values involved are very small. Also, since no packets are generated in the silence state, it makes the measurement parameters even smaller. By adopting the derivation of the model in [Yang 92] which assumes a Poisson batch arrival process (or compound Poisson process), equations for the mean and variance can be derived. The batches are defined as groups of packets which may have randomly varying sizes. The interarrival times were assumed to be independent and identically distributed. It is also assumed that the distribution of the burst lengths or the batch sizes remains constant. This means that the traffic will increase as the number of users in a session increases, but the characteristics of the traffic generated by each user or monitored at any one port, will remain constant[Forys 90].

If x is a random variable representing the interarrival time, Q. Yang et. al. [Yang 92] showed that the mean of the interarrival distribution, $E[x]$ is given by

$$E[x] = \frac{\lambda_b}{E[N_b]} \tag{4.1}$$

and it's second moment given by the following equation

$$E[x^2] = 2\frac{\lambda_b^2}{E[N_b]} \tag{4.2}$$

where $E[N_b]$ is the mean batch size and λ_b the mean batch interarrival time of a batch arrival process. N_b is the number of packets generated during a talk state.

4.1.2 Model Variables

Figure 4.2 represents the assumed model for the arrival process of the WS data stream. In the analysis, a packet refers to the number of bytes that were transferred for the application program by the operating system, during any one operating system call request. The data stream model therefore looks at the interface between the application layer and the operating system. The mapping of this interface to the actual network is discussed in section 4.4.

The definition of a burst used in the model, was as a group of packets in which the period of inactivity between any two consecutive packets is smaller than half the width of one such packet[Jackson 70]. Looking at the packet length distribution of the application programs investigated, the average packet length for all the transferred data was found to be 715 bytes with the nearest observed packet length being 1048 bytes with average effective transmission time of 0.084172 seconds. The definition of a burst was based on the average effective time it took to transmit this 1048 byte packet. Thus the threshold value was taken as 42.086ms (half the transmission time) in the following analysis. The burst was then defined as a group of packets in which the period of inactivity between any two adjacent packets is smaller than 0.042086 seconds.

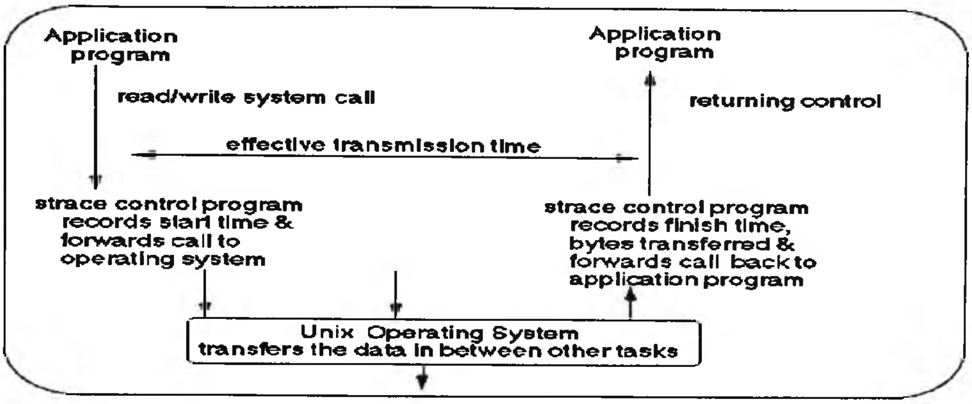

Figure 4.1: *Transmission duration : indicating the effective transmission time*

Also, for our purposes the transmission time referred to the effective time it took the control program, *strace*, to detect the onset of a request for a system call to the time the operating system returned control to the *strace* program after carrying out the request. It was found, subjectively, that the execution speed of the programs run under strace was lower than without the control program. This is caused by the extra time required for *strace* to collect and output the necessary information above the running of the application.

Adopting similar treatment to that used in [Jackson 70], the model in figure 4.2 was characterized using the following variables : -

1. Data arrival distribution in time
2. the number of packets per arrival burst segment and the number of packets per departure burst segment. These are the burst lengths.
3. packet sizes i.e. the number of bytes per departure packet and the number

of bytes per arrival packet.

4. the arrival inter-packet time represents the difference between the time at the end of the receipt of one arrival packet and the time at the start of the next arrival packet. Where the time difference is smaller than the threshold (42.086ms above) it is defined as the packet interarrival time interval. The burst interarrival time interval is the interval where the time difference is greater than the threshold.

   The departure inter-packet time represents the time difference as defined for the arrival packets, but between two consecutive departure traffic packets. The burst interdeparture time interval is similarly defined.

5. The transmission intensity for a packet is the ratio of the packet size to the effective transmission time. The burst intensity is the ratio of the sum of packet sizes of all the packets in that traffic burst to the duration of that traffic burst.

6. user response time is the time between the end of an arrival traffic burst and the start of an adjacent departure traffic burst. This marks the user's response time to the requests received from the session, or may indicate the start of a user's request.

7. idle time is the time between the end of the departure traffic burst and the start of an adjacent arrival traffic. Idle time can represent the session's

response time to the user's transmitted requests, a display update event or the start of another user's response.

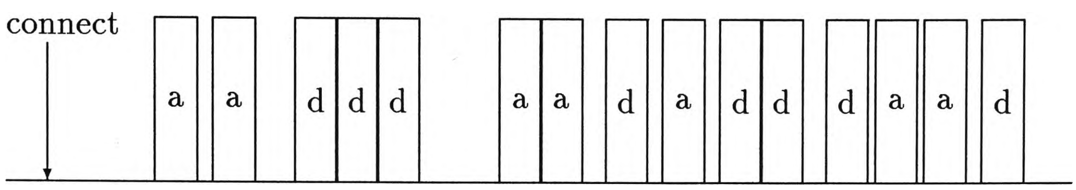

Figure 4.2: System data model:

a is the arrival packet, d the departure packet, and connect marks the point of connection

The idle time and the user response times constitute the inter-burst time intervals. From the above definitions, the inter-packet time regardless of source would be a combination of the idle and user response times, which describe the time between packets from different sources, and the packet interarrival and interdeparture times, each of which refers to traffic from the same source.

Note that these random variables are dependent on the users involved. In particular, the number of packets per arrival burst (burst length) will be very large if the users trade images, as occurs when running slide shows or dumping images on the common window. The time intervals between data transfers will increase with network loading, as the system takes longer to obtain idle conditions on ethernet.

4.1.3 Parameter choices

The following parameters were chosen to describe the random variables.

- mean value showing the central value of the distribution.
- variance which indicates the spread of the values from the mean.
- squared coefficient of variation, used in relation to the interarrival time intervals, to describe the burstiness of the traffic arrival process.

Mean

The mean represents the average value of the random variables, x_i and is calculated experimentally as x_i weighted by the corresponding probability density value. For example in interarrival time data, if N_i represents the number of arrivals within interval x_i, then the mean is defined as in equation 4.3.

$$m_x = \frac{\text{sum of observed values}}{N} \tag{4.3}$$

$$= \frac{N_1 x_1 + N_2 x_2 + \cdots + N_i x_i}{N}$$

$$= \sum_{i=1}^{K} x_i \frac{N_i}{N} \tag{4.4}$$

$$= \sum_{i=1}^{K} x_i P_X(x_i) \tag{4.5}$$

In equations 4.4 and 4.5 N is the total number of arrivals, K the number of time intervals and $P_X(x_i)$ is the probability density. Equation 4.5 is found by letting N tend to infinity in the frequency formula, N_i/N, so that the frequency becomes

$$P(X = x_i) = P_X(x_i)$$

Variance

A measurement of how much an individual observed value deviates from another gives information on the spread of the sample. The standard deviation measures the spread of the observed values, in relation to the mean value. It is found as the positive square root of the variance, σ^2. For a whole population, the variance is obtained by first squaring the deviations from mean, and then averaging the squares, an equivalent of a mathematical expectation $E(x)$ of the random variable $(X-\mu)^2$ measured in squared units. This is represented by the following equations

$$\begin{aligned}
\sigma^2 &= E[(X-\mu)^2] \\
&= \sum_i^N (x_i - \mu)^2 P_X(x_i) \\
&= \frac{1}{N}\sum_i^N (x_i - \mu)^2 \qquad (4.6)
\end{aligned}$$

Variance therefore measures the spread as the average of the deviation of the random variable x from the mean. x_i is the i^{th} observation and N is the population size. The larger the variability the less predictable the statistic is. The variance of the sum of the random variables can also be calculated using the following

$$\mathrm{var}(X) = E[X^2] - (E[X])^2 \qquad (4.7)$$

with $\mathrm{var}(x)$ representing the variance of random variable x. Hence for vector X

with x_i, $i = 1, 2, 3, \cdots$

$$\mathrm{var}(x_1 + x_2 + \cdots) = \mathrm{var}(x_1) + \mathrm{var}(x_2) + \cdots - 2\mathrm{cov}(x_1 x_2) - \cdots \tag{4.8}$$

where $\mathrm{cov}(x_1 x_2)$ refers to the autocovariance between random variables x_1 and x_2 of the vector X.

For independent random variables, the covariance terms in equation 4.8 will be zero making the variance of the sum of independent random variables equal to the sum of the variance of each random variable. The variance grows as more random variables are added to the sum, because there will be no cancelling effect occurring. It is possible to use this effect to test for independence between random variables. The derivations given so far, for the variance, refer to the situation where a whole statistical population is considered. For a sample of the data, the sample variance s^2 is calculated in the same way but by substituting the sample mean, x_m, in place of μ and the value $(sample\, size - 1)$, in place of the population size.

$$s^2 = \frac{1}{N-1} \sum_i^N (x_i - x_m)^2 \tag{4.9}$$

Squared Coefficient of variation

Also referred to as the index of dispersion (IDI) [Habib 92], [Sriram 86], the squared coefficient of variation is used to look at the relationships among successive interarrival times. It has been shown [Sriram 86] that the complexity of the arrival process is represented by a high value of burstiness. Unlike the original definition, the k^{th} sequence of interarrival times is defined as $\{X_{ik},\ i \geq 1\}$, a stationary process, then the sum of the interarrival times, $S_k = X_{1k} + X_{2k} + X_{3k} + \cdots + X_{ik}$ for i consecutive interarrival time intervals in any k^{th} sequence. Then the squared coefficient of variation or IDI is

$$\text{IDI} = \{c_k^2,\ k \geq 1\} \tag{4.10}$$

with

$$\begin{aligned} c_k^2 &= \frac{k\sigma_{S_k}}{(E[S_k])^2} \\ &= \frac{k\sigma_{Sk}}{(kE[X_{1k}])^2} \qquad (4.11) \\ &= \frac{\sigma_{Sk}}{k(E[X_{1k}])^2} \qquad (4.12) \end{aligned}$$

The square of the mean serves to normalize the cumulative covariances. Normalization makes the SCOV for the poisson process equal 1. This fact is used in defining the burstiness of a process, where the deviation of a process' coefficient from that of the poisson process, measures the degree of burstiness. The denominator in equation 4.11 above is generated using the assumption that the random variables X_{ik} are stationary. This gives the following relationship

$$E[X_{1k}] = E[X_{2k}] = \cdots E[X_{ik}] = kE[X_{1k}] \tag{4.13}$$

The variance of S_k can be calculated using the following equation

$$\sigma_{Sk} = \text{cov}(X_{1k}, X_{1k}) + \cdots + \text{cov}(X_{ik}, X_{ik}) + \cdots + 2\text{cov}(X_{ik}, X_{jk}) + \cdots \tag{4.14}$$

for any $j \neq i$. The covariance function, cov(), is defined in appendix D.

Using the assumption that the random variables are stationary, the following equations result :

$$\mathrm{cov}(X_{jk}, X_{jk}) = \mathrm{cov}(X_{ik}, X_{ik}) \tag{4.15}$$

and

$$\mathrm{cov}(X_{ik}, X_{(i+m)k}) = \mathrm{cov}(X_{jk}, X_{(j+m)k}) \tag{4.16}$$

for any i, j, or m so the covariance depends only on the time difference(m) between the random variables. Substituting these equations to equation 4.14, the variance of S_k simplifies to

$$\sigma_{Sk} = k\mathrm{cov}(X_{1k}, X_{1k}) + 2\sum_{j=1}^{k-1}(k-j)\mathrm{cov}(X_{1k}, X_{(1+j)k}) \tag{4.17}$$

From equation 4.17, the variance of the sum of time intervals is dependent on the autocorrelation/autocovariance functions. Therefore this variance describes the arrival process indicating how the random variables relate to each other.

In this research, the variance of S_k was calculated using the equation below adapted from [Sriram 86]

$$\sigma_{Sk} = \frac{1}{N_k}\sum_{k=1}^{N_k} S_k^2 - (\frac{1}{N_k}\sum_{k=1}^{k=N_k} S_k)^2 \tag{4.18}$$

where

$$S_k = x_{100k-99} + x_{100k-98} + \cdots + x_{100k+i-100} + \cdots + x_{100k}$$

for a sequence of interarrival times of length 100. To calculate the SCOV, equation 4.18 was then substituted in equation 4.12 with the value of the mean calculated as in section 4.1.3.

Thus c_k^2 measures the cumulative covariance over k consecutive blocks of interarrival time intervals. This models the cumulative effect of small individual covariances which may result in large packet delays. Delays in this case refer to time taken to reach the statistics collection ports. As relevant covariances depend on the traffic intensity, it follows that the IDI will depend on the traffic intensity. High values of c_k^2 indicate a high burstiness.

The length of each k^{th} sequence of interarrival times was arbitrary chosen to be 100 first, i.e. each k^{th} sequence consisting of $\{X_i,\ i \geq 1 \leq\ 100\}$ interarrival intervals. Other investigated block lengths were 50 and 3000. The resulting values of the SCOV were found to be in the same orders of magnitude, even though the lengths of blocks had increased.

4.2 Data Treatment Methods

This section describes the group of tests that were applied to the collected traffic statistics. The Chi-square goodness of fit test was used to find out how closely the frequency distributions fitted a standard function. The Log Histogram method was an optional test which could have been carried out to further verify the curve fitting. To test for independence among time interval data, the chi-square test for independence was carried out.

4.2.1 Chi-square goodness of fit tests

In searching for standard function forms, to describe the obtained data's random variables, chi-square goodness of fit tests[Strait 83] were performed. These were used to investigate how standard functions like the exponential distribution for continuous random variables and the geometric distribution for discrete random variables, described the obtained data.

The random variables used to investigate the decision rules, are referred to in the following section as the test statistics. These are the variables discussed in the previous section.

The goal of the chi-square goodness of fit test was to test the null hypothesis that the test statistics' distributions fitted a chosen standard function distribution. Below are the steps followed in the investigation.

1. Histograms were created for the parameter time series and the y axis value f_i calculated as follows depending on the parameter being considered:

    - the number of interarrival time values falling into a time interval, in the case of interarrival time intervals.
    - the number of bytes transferred at a particular time value, in the case of arrival data.
    - the average transmission intensity for a time interval, in the case of the duration of transmission.

2. Several values of the time intervals were used; for the packet parameters, six intervals lying between 100 μs to 10 milliseconds, and for the burst parameters, six intervals between 1 milliseconds and 100 milliseconds were investigated. Refer to section 4.5.1 for the criteria used to choose these intervals.

3. The parameters necessary to completely define the standard function distribution of the null hypothesis were estimated from the traffic data collected. For most standard functions, these were the mean and variance.

   Example standard function distributions investigated were

   - normal distribution

   $$e_i = \frac{\exp^{(-\frac{(x_i-\mu)^2}{2\sigma^2})}}{\sqrt{2\pi\sigma}} \tag{4.19}$$

   with mean μ and variance σ^2

   - geometric distribution

   $$e_i = q^{x_i-1}p \tag{4.20}$$

   with mean $\frac{q}{p}$ and variance $\frac{q}{p^2}$.

   Value of $x_i = 1, 2, 3, \cdots$ with $0 < p < 1$ and $q = 1 - p$

   - exponential distribution

   $$e_i = \frac{\exp^{(\frac{-x_i}{\theta})}}{\theta} \tag{4.21}$$

   with mean θ, variance θ^2 and $x_i > 0$

Note that, e_i is the expected frequency and x_i is the value of the i^{th} time interval. The mean and variance were taken from the variables' graphs.

4. With f_i as the observed frequency (i.e. the probability density function from the graphs) and e_i the expected frequency calculated from the standard function distributions given in step 2; a graph of f_i and e_i on the same time series is then plotted.

5. : χ^2 was then computed from the following ratio

$$\chi^2 = \sum_{i=1}^{m} \frac{(f_i - e_i)^2}{e_i} \tag{4.22}$$

where m is the sample size. χ^2 is then a random variable with chi-square distribution and $(m-t-1)$ degrees of freedom. t is the number of parameters estimated from the sample in order to define the chosen standard function distribution, e.g.for the normal distribution, t equals 2 counting μ and σ.

6. The value of $\chi^2_{\alpha,m-t-1}$ was looked up from tables of χ^2 distribution with α as the level of significance which indicates the error of rejecting the null hypothesis given the null hypothesis is true. The null hypothesis is rejected if

$$\chi^2 \geq \chi^2_{\alpha,m-t-1} \tag{4.23}$$

The results for the chi-square goodness of fit tests are tabulated in table 4.1, for those distributions which closely resembled the negative exponential distribution.

The values of A and B are coefficients of the fitting curve

$$e_i(x) = A * \exp^{(-1*B*x_i)} \tag{4.24}$$

which can be rearranged to give the following equation

$$e_i(x) = K * \lambda \exp^{(-\lambda * x_i)}$$

where λ is the mean value and K a scaling constant. The results show that the random variable distributions were closely described by the fitting curves. This is indicated by the small chi-square values.

4.2.2 Log Histogram method

Looking at the two functions that were nearer the shape of the data, i.e. the geometric and the negative exponential, it is possible to get a linear relationship after taking the logarithm of the distribution equation. This test is carried out to investigate the possibility of piecewise linearity in the observed data which would indicate the desirable compound Poisson arrival process or otherwise.

- For the geometric distribution, taking logarithms of the distribution function gives the equation

$$\log e_i = \log p + (x_i - 1)\log q \tag{4.25}$$

$$= x_i \log q + (\log p - \log q) \tag{4.26}$$

which is a straight line graph of gradient $\log q$

Table 4.1: Chi-square goodness of fit tests. *The chi-square values are for those random variable distributions which resemble the negative exponential distribution. A and B are the coefficients of equation 4.24 In the figure dep.=departure data, arr.=arrival data, wsa=ws arrivals, wsd=ws departures, where marked *, the curve is defined by two exponential functions. The number in brackets is the coefficient of the second exponential curve.*

variable	A	B	χ^2	deg.of freedom
burst length-audio dep.	0.4	0.4	0.8197	16
burst length-ws dep.	1.0	0.7	0.1878	16
burst length-audio arr.	0.4	0.4	0.8197	16
burst length-ws arr.	1.0	0.7	0.1193	16
Interdepartures-mike	0.2(0.1)*	8(17)*	1.6136	146
Interdepartures-ws	0.12	10	0.069	146
Interarrivals-speaker	0.3	16	0.4051	146
Interarrivals-ws	0.13	10	0.0671	146
Intensity-mike	0.3	3.5	3.7289	146
Intensity-wsa	0.01	5	1.3687	146
Intensity-speaker	0.1	2.5	6.8618	146
Intensity-wsd	0.1	10	18.405	146
Idle time-audio	0.1	13	0.2629	146
Idle time-ws	0.8	80	0.6418	146
User response time-audio	0.08	12.5	0.0828	146
User response time-ws	0.6	60	0.7726	146

Table 4.2: Contingency Table : for iid tests. The figures are taken from the packet interarrival times for the ws application

	Observed values f_i			Expected values e_i		
	set(1)	set(2)	row sum	set(1)	set(2)	row sum
	2.215	2.084	4.299	0.98901	3.30999	4.299
	23.905	91.741	115.646	26.60514	89.04086	115.646
	1.95	2.023	3.973	0.91402	3.05898	3.973
	140.756	344.708	485.464	111.68426	373.77974	485.464
	2.004	131.170	133.174	30.63757	102.53643	133.174
column sum	170.83	571.726	742.556	170.83	571.726	742.556

- For the exponential distribution, taking logarithms gives the equation

$$log_e e_i \quad = \quad -\log_e \theta - \frac{x}{theta} \tag{4.27}$$

again a straight line but with gradient $-1/\theta$

Thus a log histogram with piecewise linear segments may indicate a combined Poisson process. This test was not carried out owing to the desirably low values in the chi-square goodness of fit tests.

4.2.3 Independence (iid) tests

To establish if a sequence of random variables (interarrival times in this instance) are independent generally requires more than just testing the independence of consecutive pairs. The IDI for example would be ideal for this purpose [Sriram 86]. However, due to time and processing constraints we were only able to test the independence of pairs of variables in this thesis. The chi-square test for independence was used [Lapin 90]. iid refers to independent and identically distributed random variables. Below are the steps followed to perform the test

1. The chi-square statistic was used to test the null hypothesis that two sets of random variables were independent.

2. A significance level was then selected for this decision rule. This value, as in the goodness of fit tests, gives the *probability of miss*, the chances of rejecting the null hypothesis given that it is true.

3. The two sets of the random variables were then set up in a contingency table (Table 4.2) which uses figures taken from the inerarrival time intervals for the *ws* program. From this table,

   - set(1) = {2.215, 23.905, 1.95, 140.756, 2.004} and
   - set(2) = {2.084, 91.741, 2.023, 344.708, 131.170}

   The null hypothesis then was that the value of a random variable in one set was independent of which set was chosen. These observed values were denoted as f_i. The sets were first chosen as follows:

   set(1) consisted of the first, third, fifth, etc i.e. odd numbered observations

   set(2) consisted of the second, fourth, sixth etc i.e. even numbered observations

   Then the sets were chosen as groups of non-overlapping intervals.

4. The expected values of the random variables if the null hypothesis was true were calculated by keeping the totals of the columns and rows in the contingency table constant. The expected random variables were then calculated as the product of the row sum and the column sum divided by the total sum of all the rows and columns.

   For instance, the expected value of row 1 and set(1) is

$$e_{11} = 2.299 * \frac{170.83}{742.556}$$

5. The random variable χ^2, representing the test for independence, was calculated as

$$\chi^2 = \frac{1}{\text{total row sum}} \sum_i \sum_j \frac{(f_{ij} - e_{ij})^2}{e_{ij}} \qquad (4.28)$$

6. A decision was made after looking up the chi-square value in the tables using the chosen level of significance (step 2 above) and the number of degrees of freedom. The number of degrees of freedom are calculated as

$$(\text{number of rows} - 1)(\text{number of columns} - 1)$$

   The hypothesis was rejected if the calculated chi-square value was the greater of the two. In that case it meant that the random variables were not independent.

On applying the test, the results indicated that the packet interarrival time intervals were not independant of each other. The calculated chi-square values are

Table 4.3: Chi-square test for independence.

The chi-square values are for groups of non-overlapping time intervals, averaged over all the runs.

threshold χ^2 value 43.773 at significance level 0.05			
random variable	χ^2 for bursts	χ^2 for packets	degrees of freedom
Interdeparture time - ws	1.2535	127.69	32
Interarrival time - ws	7.806	158.22	32
Interdeparture time - audio	1.2148	4891.44	32
Interarrival time - audio	3.1519	23.56	32
Interdeparture time - ws/audio	5500	~26000	32
Interarrival time - ws/audio	1322	~70000	32

displayed in table 4.3. This figure also shows that the burst interarrival time intervals were independent of each other for separately considered audio and ws data traffic.

4.3 System Environment

The application programs were executed on an Ethernet network, at the University of Wollongong's electrical and computer engineering(ECE) department. Figure 4.3 gives a simplified layout of the network configuration. The network operates at 10Mbits/s connecting over 6000 computers, with about a 100 of them in the ECE. The network carries IP, Novell and ethertalk traffic. The experiments were performed using TCP/IP and UDP/IP protocol stacks.

Two configurations of the Collaborative Work system were used in the collection of data.

The first configuration involved computer terminals on the same network(elec.uow.edu.au) carrying normal load i.e. about 1 Mb/s on average as determined by displaying the ethernet traffic. These terminals were located in the Postgraduate Computer Laboratory(35G46).

Next the sessions were run between terminals again on the same LAN but located in different Labs with a round trip delay of 2ms (i.e. between the 4^{th} Year Thesis Lab and 35G46). A session in this project refers to the execution of the conference application programs described in Chapter 3.

The second configuration involved workstations on different local area networks. The networks involved were the computer science department's (cs.uow.edu.au)LAN and the Switched Networks Research Centre's LAN(snrc.uow.edu.au). The route to these networks is shown in figure 4.3.

Both two and three terminals per session configurations were used. Participants were volunteers who were given roles to play whilst the control program recorded the statistics. A variety of situations were enacted including :

1. a meeting between directors discussing the management structure of their new company.

2. architects discussing building plans.

3. telecommunications engineers planning for a local area network.

4. A lecturer and his student discussing a research topic.

Role-play was chosen because it gave us control on when the sessions took place and on the topics discussed. That way it was possible to analyse data arising from a variety of situations. For instance, scenerio 1 would generate more collaborative data the audio due to a lot of sketching, whereas scenerio 4 above may result in more audio data transfer in one direction than the other.

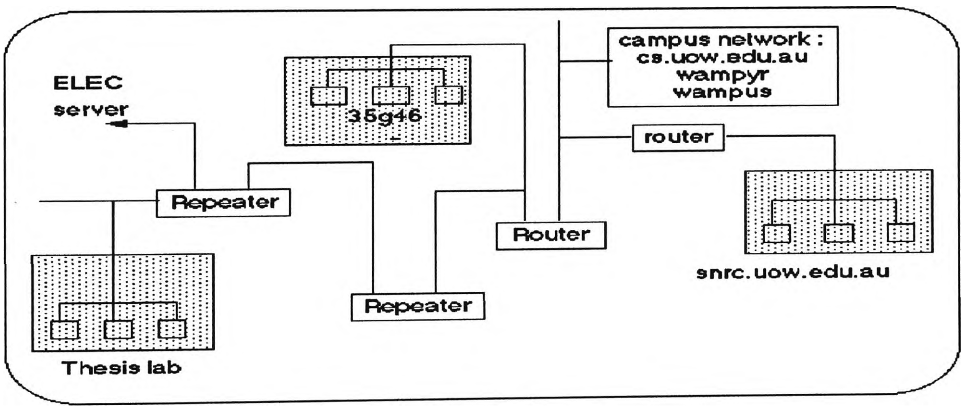

Figure 4.3: The Local Area Network diagram

4.4 Data collection

Data was collected at the ports of the WS server terminal through the use of a control program that traced the system calls occurring during the running of the collaborative work system.

Usually programs monitoring data look at the packets passing through the network and extract the address and packet length information. In this investigation, statistics were collected at the application layer interface. In the case of the transmitted data, statistics were collected before the data was encapsulated with the address and header information. For the case of the received data, this was after the data had been demultiplexed or stripped of the packet header information. Figure 4.4 shows the structure of the data collection mechanism. Our concern was for traffic profiles generated by our particular user process. By modelling it at the operating system interface, it would then be independent of the layers below as long as they can supply the required characteristics at the interface to the user process layer. It would then be possible to substitute the lower layers by any network protocol stack and expect to observe the same application behaviour. A brief description of the Unix operating system is given below.

4.4.1 Unix networking

Unix supports TCP/IP which is accessed by specific system calls. For instance, *connect* initiates a connection with a remote socket, *send* sends a message through a given socket and *recv* receives a message on a given socket[Tanenbaum 88]. The sockets in Unix are the end points to which connections can be made from the operating system and to which processes can be attached from the user applications layer. When a receiving socket is created, it is bound to a name and allocated buffer space for storing incoming connection requests. To receive a request, a new socket is created and used for that connection, leaving the first, (*the bound*) socket free to receive more connection requests. The TCP layer accepts long packets from the user process and breaks them down into datagrams to be transmitted through the IP layer. The TCP layer is responsible for time out, retransmissions, reassembly and error control of the transmitted datagrams. The IP layer provides connectionless service and attaches header information to the datagrams before forwarding them to ethernet.

The control program used in collecting data was a modified version of the strace program from Erasmus University Rotterdam. By configuring this program to time-stamp each system call at the start, and at the end of the system call, it was possible to extract the received and the transmitted data information, including the effective duration of transmission. These time values and the amount of data units sent, were then used to calculate data transmission variables like the intensity of transmission or the interarrival time intervals.

A total of 9 runs were carried out during September 1993 resulting in a total of 4 hours session time and 171529 packets handled. From this data, the series of

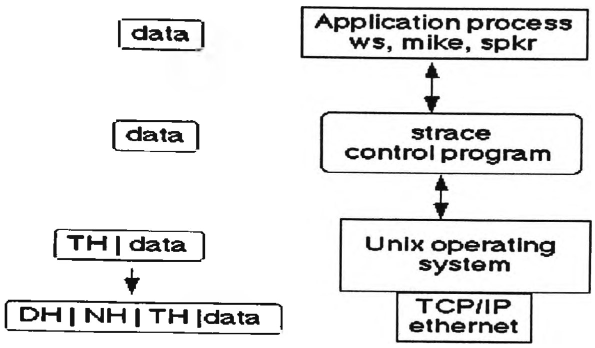

Figure 4.4: Strace: control program for data collection

values of the random variables of the assumed model were calculated.

4.5 Data treatment

In the analyses, raw data was arranged in a series of adjacent blocks or *class intervals* [Lapin 90] with the class frequency calculated as the number of observations of a particular variable falling into that class interval, e.g. frequency of occurrence of the interarrival time intervals. Histograms were drawn with the class frequency on the y-axis and the time intervals on the x-axis. Impulses of height equal to the class frequency were used in the histograms.

Figure 4.8 is an example of a histogram relating the relative frequency to the number of packets per each burst. For example, from that graph, the percentage of traffic bursts found with a length of 2 packets was 20% for ws arrival data and 25% for the audio arrival data.

Attempts were then made to categorize the obtained shape of the frequency distribution with known standard distributions like the geometrical and the negative exponential. The assumption was that the rough shape resulting from this sample data would be an estimate of the smoother frequency curve that might result if an infinitely large sample was used.

4.5.1 Choosing interval width

As the determination of the width of time intervals is crucial to the description of the frequency distribution, several intervals were tested, namely the following - 100μs, 5ms, 50ms, 100ms, and 500ms. The choice was then based on the compromise between the interval having too much detail or too little as to be detrimental to the shape of the distribution function. Too small an interval (100μs width) resulted in jagged, difficult to describe plots and an interval width greater than 0.2 seconds tended to miss important frequency pattern information. The interval giving the most distinct shape of the frequency was found at 1ms interval width for the packet distribution and 10ms width for the burst segment distribution. This result is intuitive as the bursts behaviour of the traffic is at a greater time scale than the packet traffic. Equal length interval widths were adopted for simpler interpretation.

4.5.2 Relative frequencies

In order to easily adapt the frequency curves to traffic data, of different sample sizes, and which may have been obtained under different conditions, relative frequencies were used in all the graphs. These were calculated by dividing each class frequency by the total sample size (- the sum of all the class frequencies). The graphs would then display the relative frequency distribution or the probability density function. This presentation makes it easier to compare the graph characteristics with other similar applications. Real quantities do not convey the plotted value's relationship to the total. For instance, knowing that 50% of the data is transferred within 0.02s conveys a lot more information than being told that 5000 bytes are transferred within 0.02s.

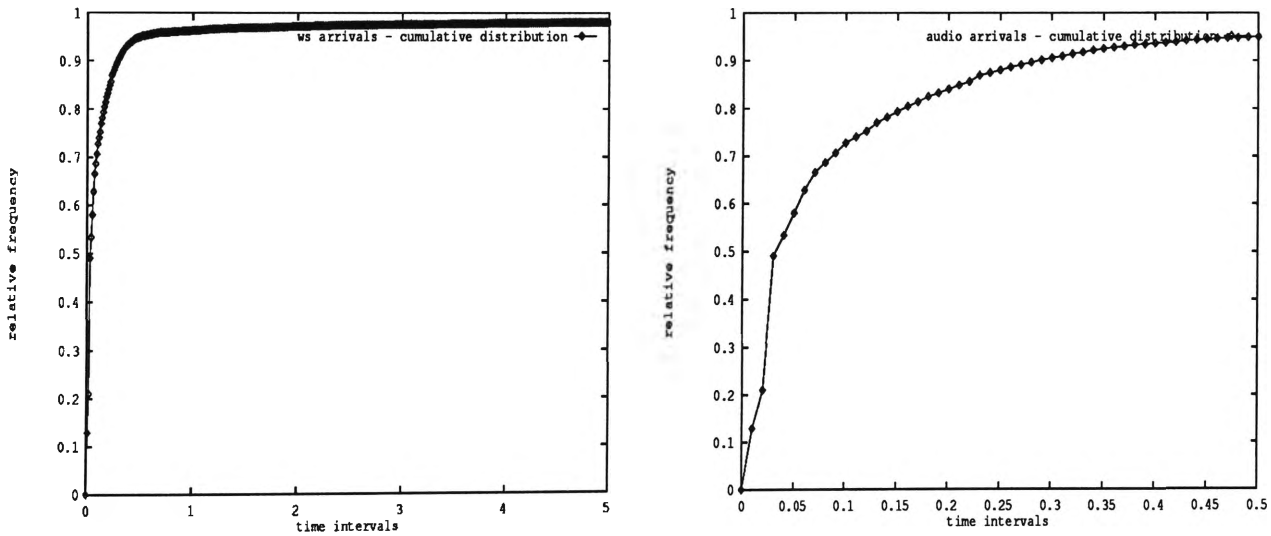

Figure 4.5: Cumulative frequency distribution to find the time interval upper limit

4.5.3 The upper limit

To find the upper limit for the time intervals, a cumulative frequency distribution (similar to figure 4.5) was used with the x - axis time intervals stretching up to the highest calculated time interval (i.e highest observed value). This was done by adding each class frequency to the sum of the lower class frequencies, hence giving information on the level of the class intervals. The upper limit was then taken to be the interval below which 95% of the frequency data falls. The upper limit gives the limit beyond which the obtained data no longer behaves typically and is an insignificant proportion of the data.

4.5.4 Confidence Interval

The confidence interval defines the precision and the reliability of the obtained data. Reliability refers to the probability that the estimate is correct; and precision is the probability that the estimate is close to the target parameter. In the random variables investigated, both the y-min/y-max values and the standard deviation were used as confidence intervals. The y-min and y-max values were found from the averaging of the histograms obtained for each test session run. The standard deviation value was calculated as in the definition given in section 4.1.3.

Standard distribution functions to fit the variables of the traffic model were then investigated from this data.

4.6 Data Analysis

This section reports results obtained during session experiments, and discusses the interpretation of the results in terms of the analysed variables.

4.6.1 Arrival packets

Figure 4.6 shows graphs of the distribution of the arrival and departure processes. The graphs indicate the similarities of the obtained profiles with the assumed data model of section 4.1.2. As in the model of figure 4.2, the graphs show multiple arrivals and departures confirming the theory of batch arrivals (burst segments). They also show burst segments in which the packets within the segment have different lengths (-randomly varying packet sizes). Note that the inter-packet times and the inter-burst times have varying lengths. These intervals are discussed in detail in section 4.6.4.

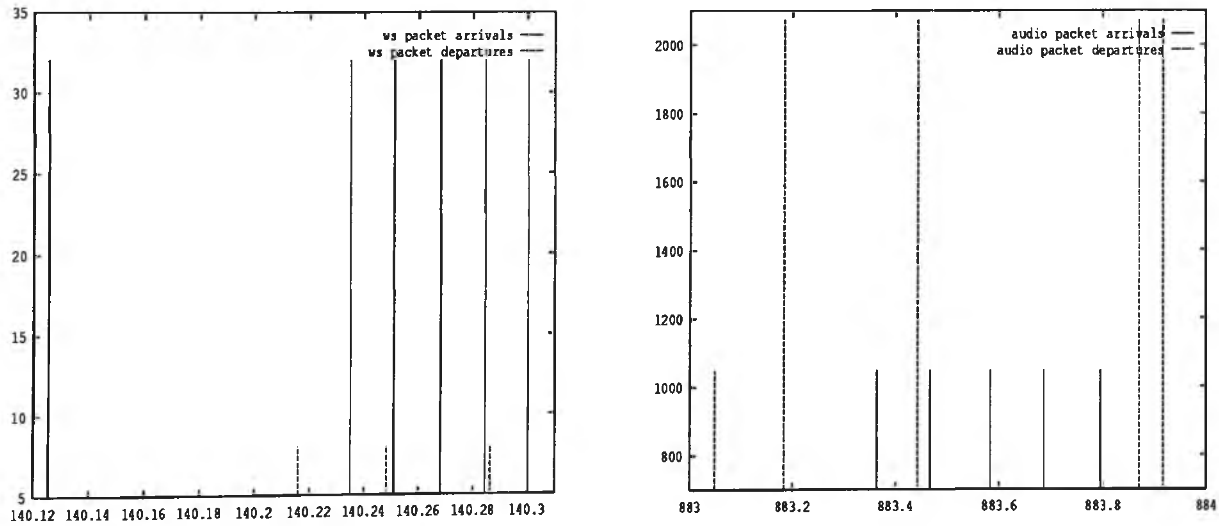

Figure 4.6: Comparison between arrival and transmitted data

The graphs in figure 4.7 shows that most of the data handled has packet lengths that are less than 5000 bytes with only a few having the maximum of 8000 bytes length. There is therefore a possibility of trading-in the bandwidth of the transmission equipment, to a little degradation in system performance. This could be done by allowing a bandwidth of just half that of the peak traffic handled. A possible situation might be the running of two sessions where only one session was fully allowed for. This assumes that the two sessions do not carry peak traffic at the same time. If they do, the result may be an increase in waiting or response times, as the processes wait their turn to transmit.

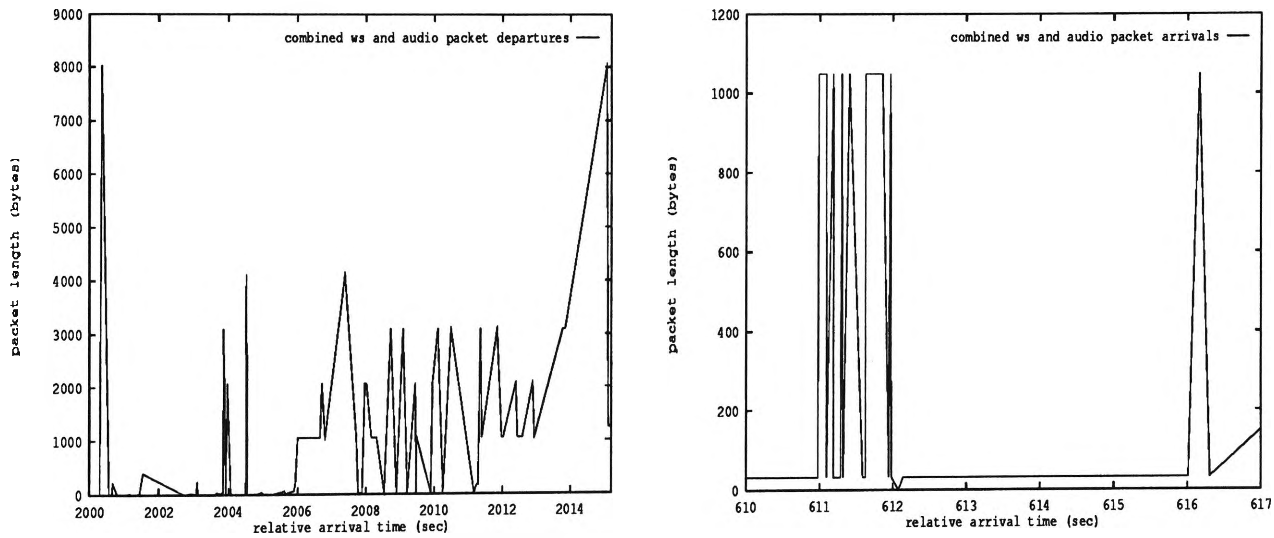

Figure 4.7: Typical transfer data: combined ws and mike

Similar traffic patterns were obtained for both the sessions run within the same local area network and those run across separate local area networks. The data collected from the two configurations, was therefore combined in the analysis to improve the confidence intervals.

Graphs for the arrival processes of the ws application and the audio communications when considered separately are displayed in the appendix (figure A.1). The graph from the audio application shows an almost constant packet length, for the arrival data collected by *speaker* program, of around 1048 bytes whilst the ws program exhibits large variations of packet length, with a minimum of 8 bytes. Comparing with the graphs in the appendix for the transmitted data, (figure A.2), it is clear that the largest volume of data is generated by the audio communication program, and thus has a greater influence on the CSCW system's overall traffic characteristics.

4.6.2 Burst length

The burst length refers to the number of packets within each burst segment. This value was obtained by keeping track of the number of adjacent interarrival time intervals whose value was less than the threshold of 42.086 ms. Whenever the calculated intervals were greater than the threshold value, this was recorded as the start of a new burst segment and hence the end of the previous burst segment. The results, displayed in figure 4.8, show the burst length calculated using arrival data from the audio and from the *ws* application. Graphs from the departure data for each application are shown in the appendix (figure A.3).

From the graphs in figure 4.8, the number of packets per burst segment for both received and transmitted data showed an exponential distribution with a mean

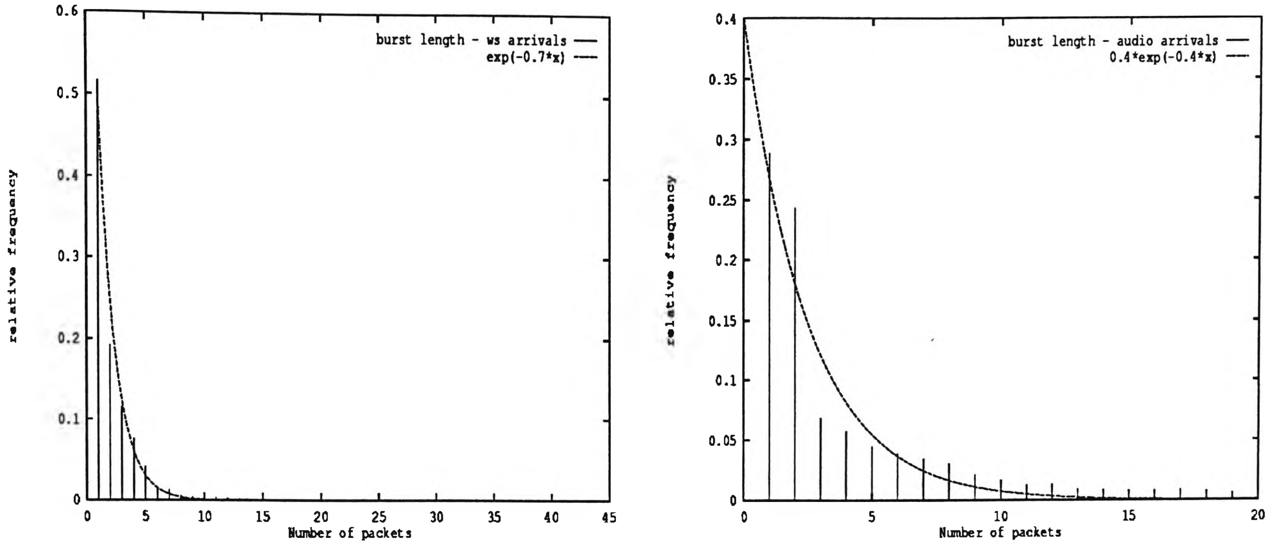

Figure 4.8: Burst Length : arrival data for ws and audio processes

value (parameter θ) of 1.429(=1/0.7) for ws data and 2.5 (=1/0.4) for audio.

The negative exponential curve, from figure 4.8, is given by the equation

$$f(x_i) = \exp(-x_i/\theta)$$

$$= \exp(-0.7x_i)$$

$$(\text{valid for } 1 <= x_i <= 12) \qquad (4.29)$$

the integral of this empirical density function is approximated by the sum of the function between given limits with interval 1, i.e., $\sum f(x_i) = 0.988$.

Hence the mean is 1.429 packets per burst segment for WS collaborative work system and 2.5 packets per traffic burst for the audio communication programs. A small mean value indicates that it is difficult to predict when the next packet is going to come given that one packet has already been transferred.

Extremes, like 12 packets per burst segment, occurred when images were transferred between displays, and in some cases, in the audio application programs.

4.6.3 Packet sizes

Another important parameter in network design and analysis is the distribution of the size of the packets handled. By looking at the number of packets of the same size, the average number of data units per packet was found. When the audio system was considered on it's own (figure 4.9), almost 99% of the packets had a packet size of 1048 bytes, with some received audio packets deviating to 8000 data units in extreme cases. For the *WS* data system, the packet size distribution for arrival data showed about 95% of packets with a packet length of 32 bytes. This brought the average packet size to about 100 bytes per packet because of the extremes of around 4000 bytes. The data displayed in the appendix (figure A.5) gives the distribution of the packet sizes for the combined application programs. The graphs show two prominent peaks, one at 32 bytes and another at 1048 bytes. This graph (figure A.5) indicates a random distribution. The first peak in figure A.5 therefore indicates the activities of the cursor movement in the *WS* application. The peak at 1048 bytes in the same graph is influenced by the audio communication. This raises the overall mean of the packet sizes to nearly 800 bytes.

The high relative frequency values in figure 4.9 suggest that the arrival process may be deterministic, consisting of only two packet sizes, 32 bytes and 1048 bytes. These values are vital in simulation work where they can be used to characterise the size and distribution of arrival packets. Another application of this parameter

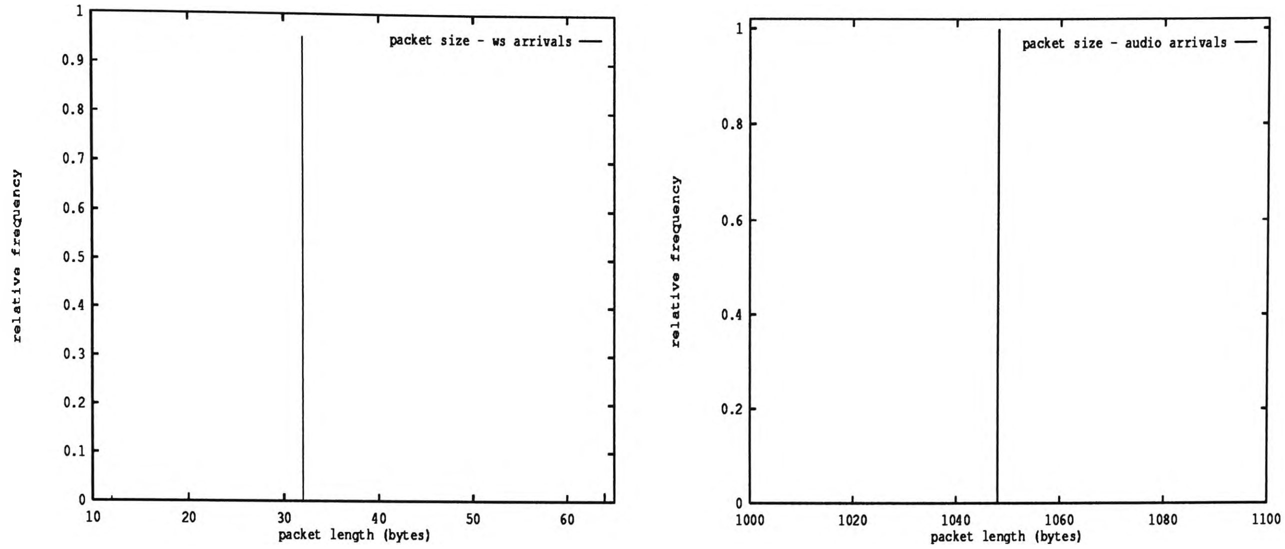

Figure 4.9: Packet size Distribution : arrival data ws and audio processes

would be to compare the largest sized packets with the average packet size, thus get a feel of how effectively the CSCW system utilizes the network. This can then be used to make decisions on how to design and dimension the network.

4.6.4 Interarrival intervals

The sequence of interarrival times is a superposition of the packet processing and transmission times [Forys 90]. As in the previous analyses, raw data was arranged in a series of adjacent class intervals with the class frequency calculated as the number of interarrival time intervals falling into that interval. The interval widths used were 1ms for the packet interarrival distribution and 10 ms for the burst interarrival distribution. These values, determined empirically, were found to give a better and more distinct distribution than the other investigated values (section 4.5.1). In network analysers, the interarrival times, together with service time data parameters, are used to estimate network response times [Forys 90].

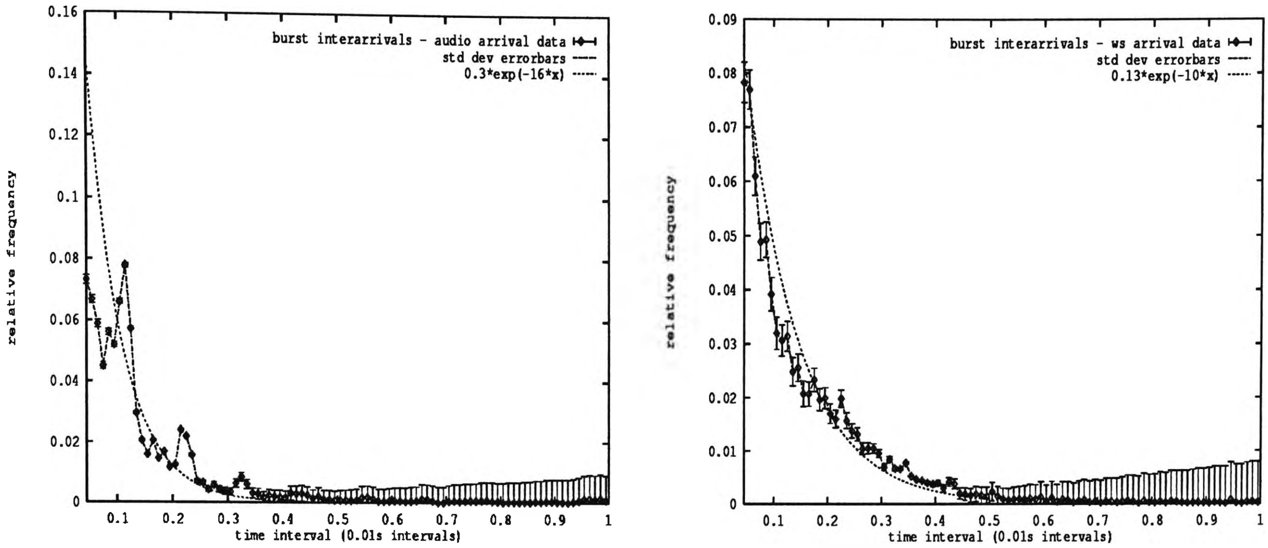

Figure 4.10: Burst Interarrival time distribution *for audio and ws processes showing best fit negative exponential curve and standard deviation confidence intervals.*

Burst interarrival times

For the burst interarrival time intervals, the vector of the time intervals was found as the length of time from the completion time of the last packet of one burst to the start of the first packet of the next burst segment; or simply the interarrival time intervals which were greater than the threshold value of 42.086 ms. The latter calculation method was used in this analysis.

The mean and variance for the interarrival time intervals were calculated and the chi-square goodness of fit tests showed an approximation towards the negative exponential distribution. The graphs, figures 4.10 and 4.11, show a mean of between 0.1 seconds (1/10) and 0.0625 seconds (1/16) for *ws* and for the audio traffic respectively. This suggests that the burst segments arrive within 62.5 ms to 100 ms of each other on average, with a variance of 0.3% to 1%. The curves from the audio program, figure 4.11 show a larger deviation from the negative exponential. The graph of the interdeparture time interval from the *mike* program indicates a possibility of being modelled as a sum of two exponential functions; one centred at zero and of mean 0.0588 seconds, and the other centred at 0.25 seconds with a mean of 0.125 seconds.

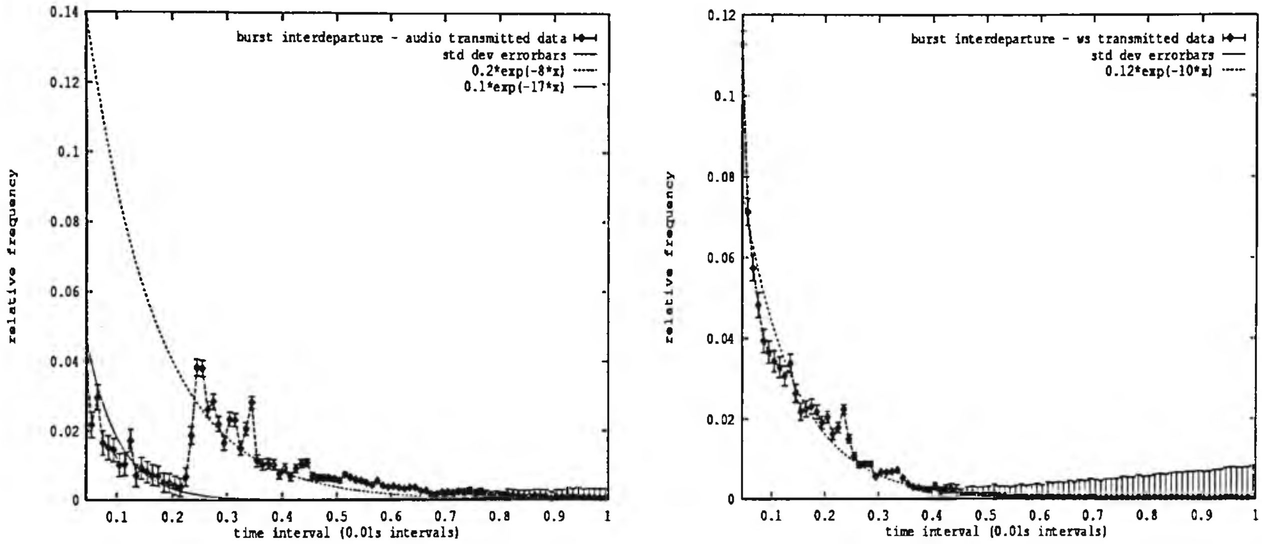

Figure 4.11: Burst Interdeparture time distribution *for audio and ws processes showing best fit negative exponential curve and standard deviation confidence intervals.*

$$f(x_i) = f_1(x_i) + f_2(x_i) \tag{4.30}$$

with

$$f_1(x_i) = 0.1 \exp^{-17x_i} \text{ for } 0 < x_i <= 0.25 \tag{4.31}$$

and

$$f_2(x_i) = 0.2 \exp^{(-8x_i)} \quad \text{(valid for } x_i > 0.25\text{)}$$

the integral of these empirical density functions is approximated by the sum of the functions between the given limits using an interval of 0.01.

$$= \quad 0.1 \exp^{(-8(x_i-0.25)}$$

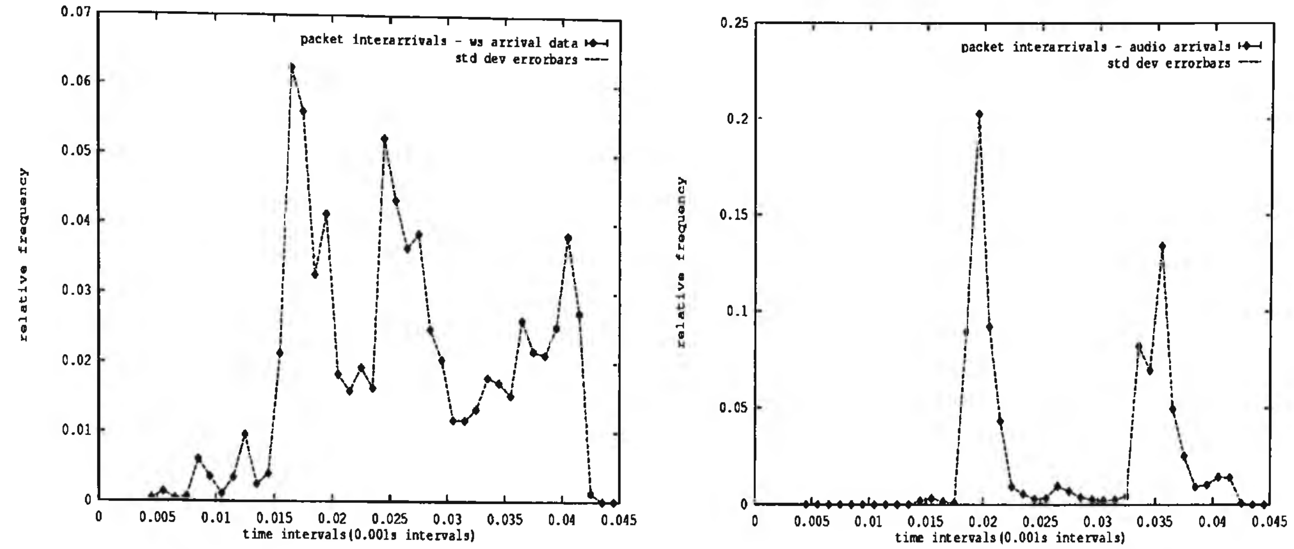

Figure 4.12: Packet Interarrival time distribution :*audio and ws processes with the standard deviation used for confidence intervals*

Packet interarrival times

To analyse the packet interarrival time distribution, a statistical treatment was used that was similar to the burst interarrival time treatment. The graph for the combined ws and audio traffic data (figure 4.13) shows two peaks, one at 0.0195 seconds and another at 0.0355 seconds, with the rest of the class frequencies scattered below the 2% value. This same pattern is indicated in the traffic profiles from the arrival audio data (figure 4.12). As in the profiles from *mike*, the distribution can be modelled as two normal distribution curves with means 0.0195 and 0.0355 seconds (equations 4.32 to 4.34).

$$f(x_i) \;=\; f_1(x_i) \;+\; f_2(x_i) \tag{4.32}$$

with

$$f_1(x_i) = \frac{\exp^{(-\frac{(x_i-0.0195)^2}{2\sigma_1^2})}}{\sqrt{2\pi\sigma_1}} \tag{4.33}$$

and

$$f_2(x_i) = \frac{\exp^{(-\frac{(x_i-0.0355)^2}{2\sigma_2^2})}}{\sqrt{2\pi\sigma_2}} \tag{4.34}$$

The interdeparture time intervals for the combined traffic data and from *mike* were different showing a high spread (almost uniform distribution) of values from about 0.01 seconds to 0.045 seconds. This trend was also evident in the traffic profiles obtained from the ws arrival and departure data.

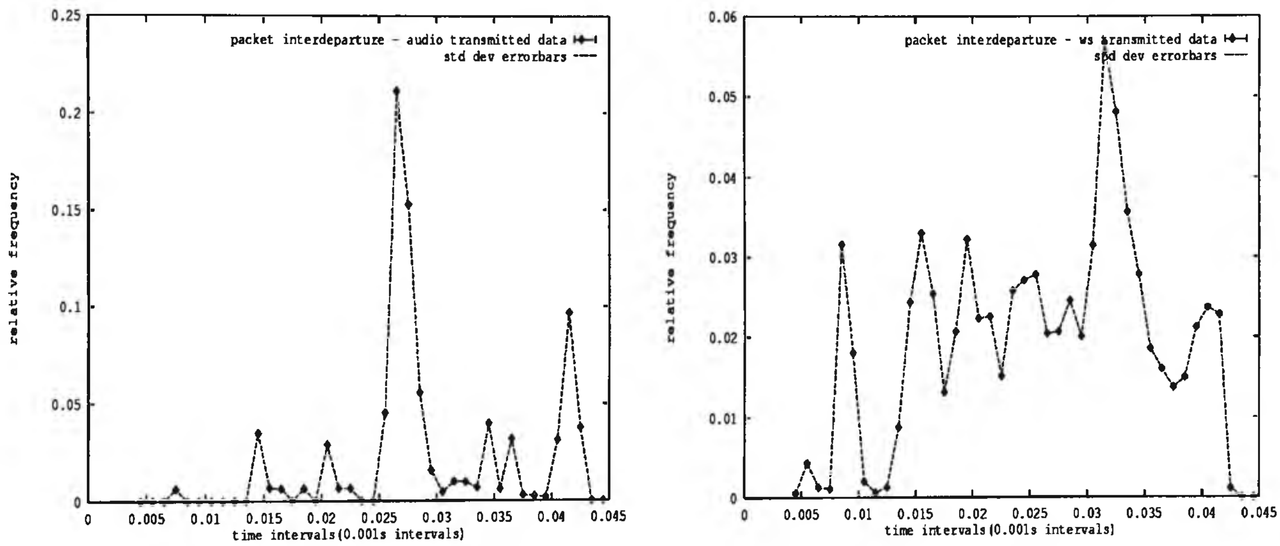

Figure 4.13: Packet Interdeparture time distribution *for audio and ws processes with the standard deviation used for confidence intervals*

The ratio of audio arrival data to ws arrival data was about 4 to 1, with a similar ratio applicable to the departure data. This could be the reason why the traffic profiles from the combined audio-*ws* programs are similar to those from the *speaker* program more than they are to the *ws* data traffic profiles. The audio departure data did not have much impact on the combined audio -ws profiles; it appeared that the peaks from the audio data (*mike* program) coincided with the dips in the ws traffic profiles because people were either talking or drawing and not doing both at the same time.

Burstiness

The interarrival time intervals indicate the dependencies between data in the arrival process. As has been discussed before, the squared coefficient of variation[Sriram 86], c_k^2, looks at the covariances among successive intervals. Hence by calculating this coefficient for the interarrival time intervals, the burstiness of the arrival data is determined. The coefficient has been defined in section 4.1.3 as

$$c_k^2 = \frac{Var[S_k]}{k(E[X_1])^2}$$

where $\{X_k\,;\,k \geq 1\}$ are the interarrival time intervals, with S_k as the sum of the interarrival intervals in the k^{th} sequence. In [Sriram 86] it is shown that c_k^2 equals c_1^2 for all k for single or less than 20 voice sources. It is not shown however whether this applies to any number of time intervals in a sample. The suggestion could mean that the whole vector of interarrival time intervals is taken as one block. When calculations of c_k^2 were carried out using k sequences of 100 and of 50 time intervals, it was found that the collaborative work system gave squared coefficient of variation figures that started very high, ($c_1^2 \approx 20000$) and approached a value($c_{55}^2 \approx 40$) lower than the initial by almost three orders of

Table 4.4: The Squared coefficient of variation for time intervals. *The sample size is given as (packets sample size):(bursts sample size)*

Arrival data							
Type of traffic	Packets			Bursts			sample size
	mean value	variance value	sqd coeff. of var.	mean value	variance value	sqd coeff. of var.	
audio	0.0246	0.0006	1.0519	0.1845	0.368	29.3757	13855:2415
data	0.0209	0.0059	13.429	0.3578	0.5005	3.9088	1516:1755
comb.	0.0210	0.0006	1.4181	0.1521	0.1651	7.1339	14149:5393
Departure data							
audio	0.0258	0.0034	5.1542	0.6091	0.1555	0.6091	2585:5553
data	0.0207	0.0037	8.6866	0.3326	0.4932	4.458	2376:1784
comb.	0.0217	0.0017	3.72	0.2652	0.1226	1.7427	5087:7212

magnitude, as k increased. In [Sriram 86], the k sequences used were of 3000 time intervals and gave the c_k^2 value as starting from 18.1 and reducing towards one as k increased. In table 4.4, this coefficient was calculated using the variance of the whole sample.

Note, the sample size is given as (packets sample size):(bursts sample size) in table 4.4; this table gives a summary of the parameters for the interarrival time intervals distributions shown in the graphs displayed in this report. The squared coefficient of variation is taken as the ratio of the overall variance to the square of the overall mean for a particular application. This assumes that the entire sample size is one k^{th} sequence of time intervals, and hence $k = 1$, giving the aquared coefficient of variation as c_1^2

The burstiness of an arrival process is important in determining the utilization of the network and in the allocation of network bandwidth. A higher burstiness indicates large variations in the arrival data and hence results in the network being under-utilized. Thus, in addition to the squared coefficient of variation, the ratio of transmission occupation time to call duration can be used to measure burstiness. The higher the value of the squared coefficient of variation, the higher the burstiness of the variable; and the lower the ratio of occupation to call duration, the higher the burstiness.

4.6.5 Intensity of Transmission

Here two dimensions of the statistical data are investigated namely, the rate of transmission and the duration of transmission. The goal is to find any relationship between the two parameters i.e. whether large volumes of data are transferred at a higher rate or that transmission rate is constant, i.e. independent of the amount of data handled. A scatter diagram, as in figure 4.14 gives an estimation of the shape of the graph, showing a decrease in the rate of transmission as the transmission time increases. In such curves, large volumes of data would be equivalent to a large duration. The intensity of transmission shows the variation of transmission speed with the volume of data sent or received by the terminal.

The construction of the histograms for the transmission intensity curves was done by adding up all the transmission intensities of those values of the transmission duration that fell into a particular class interval, and then finding their average in relation to the number of transmission duration values found. This average intensity of transmission formed the class frequency of that particular interval. The intensity is calculated using the following equation :

$$I_k = \frac{(\text{no. of bytes transferred})}{\text{duration of transmission}} \tag{4.35}$$

All the graphs in figures 4.14, 4.15 and 4.16 therefore reflect the average intensity of transmission for time intervals.

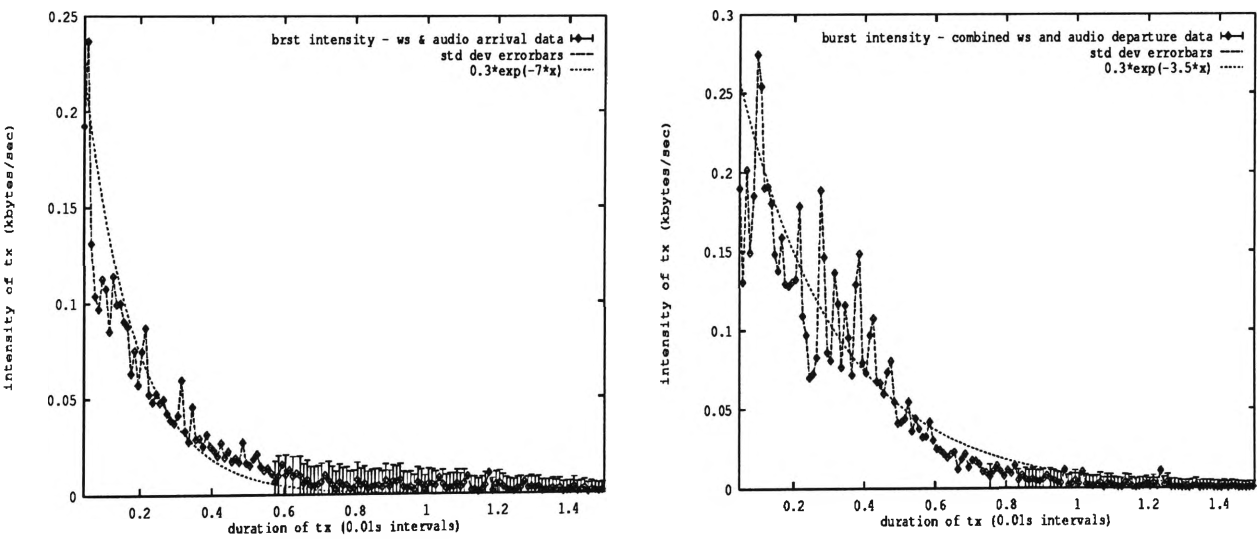

Figure 4.14: Burst Transmission Intensity : *for combined ws and audio data using the standard deviation as confidence intervals.*

Burst intensity

For the burst intensity, the number of bytes transferred per burst segment, was found as the sum of all the packet lengths of the packets within that burst segment; whilst the duration of that burst segment was the total time taken from the start of the first packet of that burst to the completion time of the last packet of the burst segment. This burst transmission duration therefore included the periods of inactivity between adjacent packet transfers, therefore results in lower transmission intensities when compared with packet intensity values. The histograms plotted for the burst intensity in figure 4.14 show good approximations to the negative exponential distributions, similar to those obtained for the burst interarrival time intervals. Figure 4.15 shows plots with large values of confidence intervals such that it is not possible to find a common distribution function to describe the data. Looking at the traffic profiles for the audio data separately from the ws system (Figure 4.15) the shape of the graphs do not show a clear exponential curve, exhibiting scattered values up to a transmission duration of around 1.5 seconds. It appears then that a lot of smoothing occurs on combination of the two applications(figure 4.14), i.e. the audio and data integration.

Some of the graphs have been drawn in Appendix A with the maximum and minimum intensity values as errorbars to indicate the range of y-axis values obtained. The fitting exponential curves have a mean value of 0.2 for the intensity of the departure burst and 0.1 for the intensity of the arrival burst. The arrival burst had

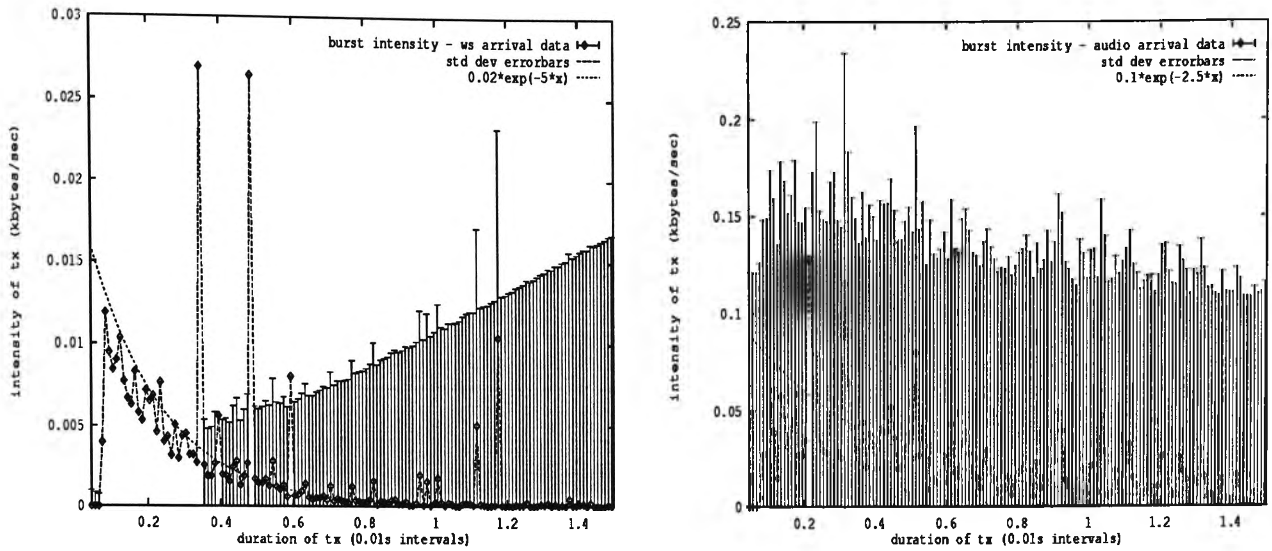

Figure 4.15: Burst Transmission Intensity : *for the arrival process of ws and audio data with the standard deviation as confidence intervals.*

a better exponential fit than the departure burst intensity, but with more data handled during the departure phase. This is more likely to be a characteristic of the session users' behaviour, i.e. which user generates the most traffic.

Packet intensity

To calculate the packet intensity, equation 4.35 was again applied, but this time to all the packets in the sample, using the number of bytes transferred as the packet lengths. The transmission duration values were calculated as the time taken from the start time of the packet's transfer to the completion time of that packet's transfer.

The plotted graphs are shown in figure 4.16. The standard distribution shape closest to these graphs is a normal distribution centred at 0.017 seconds for the profiles from ws system, and centred at 0.015 for the audio profiles. Even then,

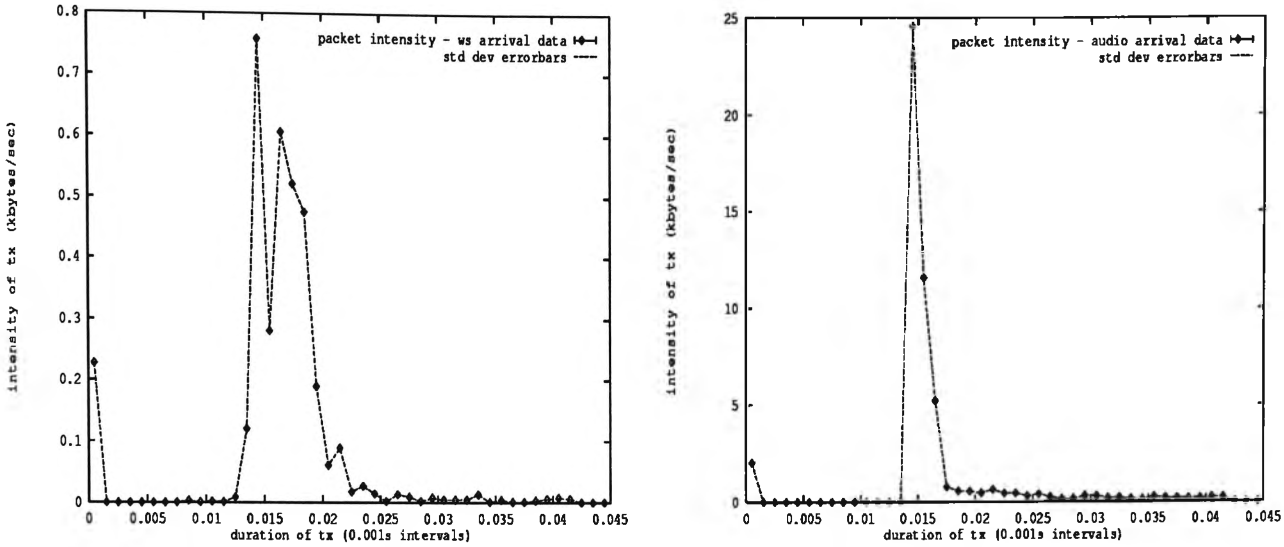

Figure 4.16: Packet Transmission Intensity : *for the arrival process of ws and audio data. Both plots have been drawn with the standard deviation as confidence intervals.*

it's a poor approximation to the shape of the graphs. The only suggestion these graphs project is that most of the packet data is transferred with a transmission duration lying between 0.014 seconds and 0.02 seconds with a packet intensity of slightly over 6 kbytes/sec. (roughly 64kbits/sec) for the audio data, and peak packet intensity for *ws* data of slightly under 1 kbyte/sec ($\approx$ 8kbits/sec). The lower data rate values could be a result of discarding potential packets due to the silence suppression employed.

4.6.6 Response times

The definitions for the idle time and the user response times have been given in section 4.1.2, where they are described as inter-burst time intervals because they look at the time intervals between adjacent packets from different directions; e.g.

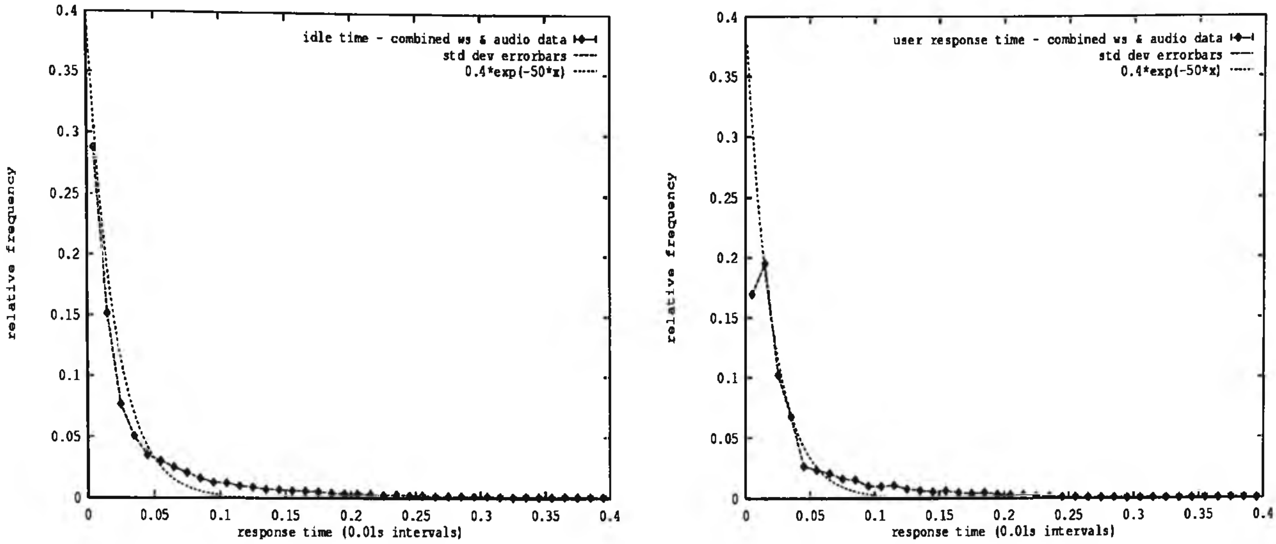

Figure 4.17: The user response time and idle time : *for the combined ws and audio data with the standard deviation as confidence intervals.*

between the last arrival and the first departure after that arrival or vice-versa. The values of these time intervals were obtained by keeping track of the traffic flow direction. The graphs for the user response times and the idle times are shown in figures 4.17, 4.18 and 4.19.

Idle times

The idle time was calculated as the inter-burst time from the end of a departure burst to the start of an adjacent arrival burst . This response time therefore consists of the time it takes the user response packet to be distributed among the session participants, and the time taken by a packet generated within the session to arrive at this same user.

The graphs shown in figures 4.17 and 4.18 show that the idle time can be

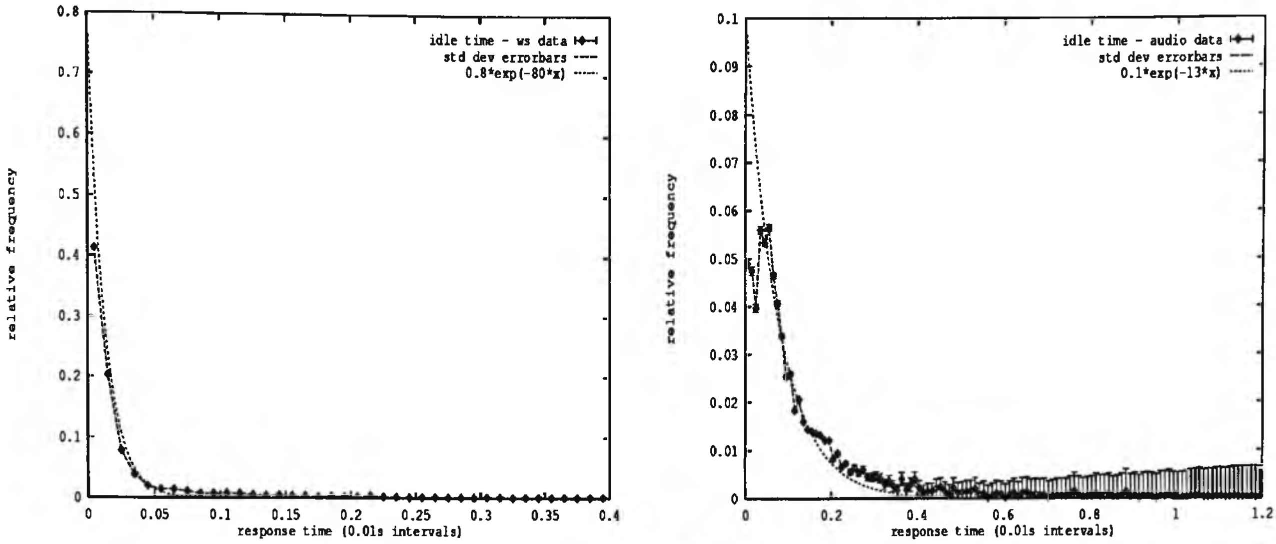

Figure 4.18: The Idle time : *for ws and audio using the standard deviation as the confidence intervals*

modelled by a negative exponential curve

$$f(x_i) \;=\; 0.4\,\text{exp}^{-50x}$$

which has a mean of 20 milliseconds. Compared to a mean of 50 to 125 milliseconds for the burst interarrival time intervals, the idle time between packets going in different directions is smaller than the inter-burst time. This shows that there is a greater chance of a departure packet being followed by an arrival packet than there is for bursts to occur. The mean value of 20 ms is close to the mean packet interarrival time of 15 ms.

User response times

The user response time was calculated as the inter-burst time from the end of an arrival burst to the start of an adjacent departure burst traffic. It thus represents

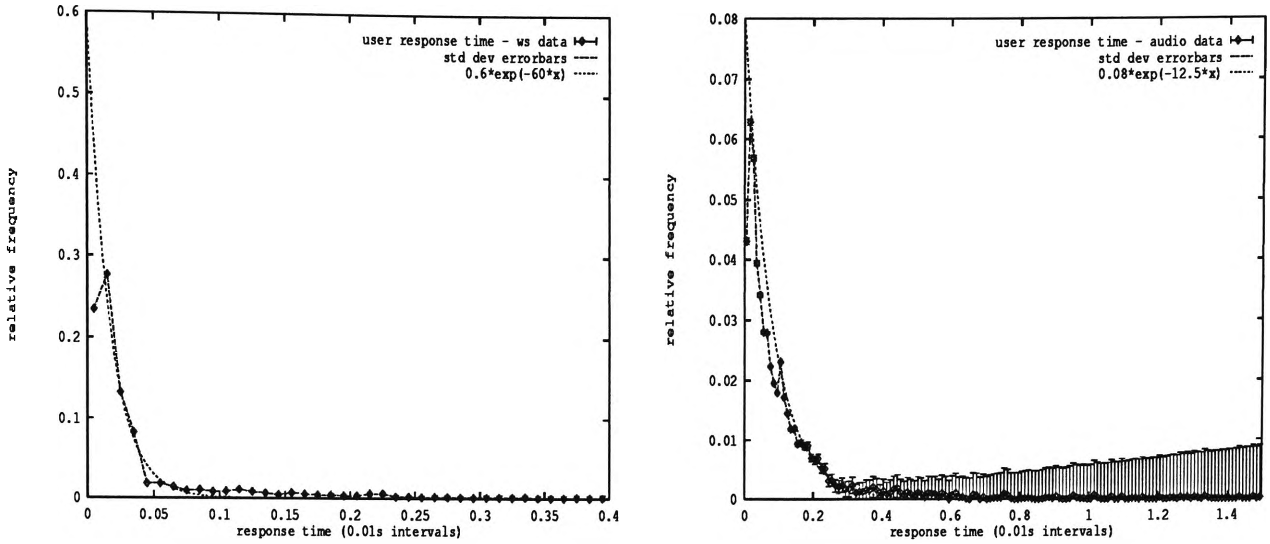

Figure 4.19: The User Response time : *for ws and audio using the standard deviation as the confidence intervals*

the time it takes the user to respond, generate a request or distribute a packet among the participants involved in the session. But because the packets come as groups the user response time is calculated starting from the end of the last packet of a group. It thus may not represent the full or actual response time of the user, but that of the application process, for instance, the data sent during polling to grab the next event.

The graphs shown in figures 4.17 and 4.19 can be modelled by the same negative exponential curve as the idle time

$$f(x_i) = 0.4 \exp^{-50x}$$

with mean 20 milliseconds.

The idle times and user response times take a larger proportion of the session holding time than the data transfer time. This relationship is affected by the number of simultaneous users on the conferencing session as well as on network traffic and the load on the users' sites. The more participants involved the greater these times become. This indicates a possibility of multiplexing this traffic, at the TCP layer for instance, for effective network utilization. The bandwidth during these periods of inactivity would then be used to carry other datagram traffic types. Multiplexing can be implemented by having more transport connections using one network socket. It can also be done by opening more than one network connection and distributing traffic from one tansport connection among the network sockets opened. The later would be ideal to improve network performance when the bandwidth is increased by the number of sockets opened. The limit on bandwidth improvement would be the capacity of the physical line[Tanenbaum 88].

4.7 Summary

Table 4.5 summarises the variables investigated giving the equation of the curve that best fits. The table shows that the random variable packet distribution for the audio data can be modelled by the normal distribution curves, whilst the the time interval distribution for *ws* are random. The analysis showed the burst intensity for the transmitted ws data with confidence intervals too large for conclusions to be made on the curve of best fit. It was also observed that when the audio and *ws* data were combined a pattern approximating the negative exponential was obtained.

Table 4.5: The Equations for curve fit

Random variable	Audio data	WS data
Pkt interarrivals	normals(0.02,0.04)	random
Pkt interdeparture	normal(0.075,0.045)	random
Pkt intensity -arr.	normal(0.015)	normal(0.0175)
Pkt intensity -dep.	normal(0.0195)	normal(0.0195)
brst interarrival	0.3exp(-16x)	0.13exp(-10x)
brst interdeparture	0.2exp(-8x)+0.1exp(-17x)	0.12exp(-10)
brst intensity-arr	random	0.02exp(-5x)
brst intensity-dep	0.3exp(-3.5x)	random
idle time	0.1exp(-13x)	0.8exp(-80x)
user response time	0.08exp(-12.5x)	0.6exp(-60x)
packet size	1048 bytes	32 bytes
burst length	0.4exp(0.4x)	exp(-0.7x)

Chapter 5

Comparison with Packet-Train Model

Past investigations into the behaviour of the data traffic transferred in computer networks have used Markov chains (reference [Sriram 86], [Fontana 89]) to describe the data arrival processes. It was also shown[Habib 92] that the different types of data sources transported on multimedia networks could each be modelled as individual Markov chains. The model presented in the previous chapter describes the statistical properties of the variables that would require specification in the Markov chain model. The distribution curves for the transmission intensity , for instance, provide for the peak and average rates, whilst the burstiness parameter describes the time-scale rate of variation. In this chapter, the model of Chapter 4 is compared to other models [Jain 86] which have been developed for computer network traffic.

5.1 Traffic model

In [Jain 86], models similar to the one developed in this research (section 4.1.2) were described but for the data link to physical layer interface. The main aim of that paper was to find out if data traffic could be grouped into correlated packets travelling together. Jain and Routhier introduced their model, for the arrival process, which consisted of trains pulling cars. Their parameters included the inter-train times influenced by the number of users transferring data, the intercar time depending on the system, and the train size which was equivalent to the data sizes. They also showed how their model could be specified to fit specific models. For example, by setting the intercar and intertrain interval distributions to be exponential, their packet train model characterized the Poisson network traffic model.

The results obtained in this research can also be applied to Jain and Routhier's model, as follows: If we identify their trains as the burst segments described in chapter 4, their inter-train intervals as the burst interarrival intervals and their inter-car time intervals as the packet interarrival time intervals (or interdeparture time intervals), table 5.1 summarizes the observed characteristics of our model. Comparing it to the models in [Jain 86], the obtained model for the combined CSCW data and audio approximates their regular train network traffic model which was given as best describing voice packets. The regular train model consists of constant packet interarrival time interval distribution and exponential burst interarrival time interval distributions. This differs from our model in the fact that our packet interarrival time distributions are not exactly constant. They exhibit a random distribution, thus describing a *random* packet- train rather than the Jain/Routhier regular packet-train. The randomness in packet distributions could be a result of the silence suppression algorithm acting on talk spurts. Further investigations are needed to verify the cause.

Table 5.1 summarises the statistical proprties of the variables investigated for comparison with the Jain/Routhier model. The tests for independence of the interarrival times showed correlation between packets. This may be the reason why the inter-packet time distributions for data are random. The inter-packet time interval distribution for audio data showed two clusters, (centered at 20ms and 40ms) approximating a normal distribution. This may be due to the fact that audio traffic is a constant rate service, transmitting most of the packets within a mean time interval. The correlation between the inter-packet time intervals is modelled by the traffic burst distributions. Table 4.2 shows that all the inter-burst time distributions are exponential. This suggests that the traffic bursts arrivals may be poisson. The intensity of transmission for traffic bursts also exhibited exponential distributions, whilst the transmission intensity for packets was normally distributed and centered at 15ms for arrival data and at 19.5ms for

Table 5.1: The Network Traffic Model

in the table, exp. refers to the negative exponential distribution curve; normal refers to the normal distribution curve; gen. refers to a general distribution; pkt. ints. distr. refers to the packet intensity distribution; burst ints. distr. refers to the burst intensity distribution; inter-arr. and inter-dep. refer to the interarrival and interdeparture time intervals; sqd coeff. of variation is the squared coefficient of variation; pkt. sizes refers to the packet size distribution

Arrival data							
Type of traffic	pkt ints distr.	burst ints distr.	pkt inter-arr. distr.	burst inter-arr. distr.	brst sqd coeff. of variat.	burst length distr.	pkt sizes mean
audio	normal	exp.	normals	exp.	29.378	exp.	1024 bytes
data	normal	exp.	gen.	exp.	3.909	exp.	32 bytes
comb.	normal.	exp.	normal.	exp.	7.134	exp.	800 bytes
Departure data							
Type of traffic	pkt ints. distr.	burst ints. distr.	pkt inter-dep. distr.	burst inter-dep. distr.	brst sqd coeff. of variation	burst length distr.	pkt sizes mean
audio	normal	exp.	normals	exp.	0.609	exp.	1024 bytes
data	normal	exp.	gen.	exp.	4.458	exp.	32 bytes
comb.	normal	exp.	gen.	exp.	1.74	exp.	800 bytes

departure data. The packet sizes for the arrival data were an almost constant size of 32 bytes for ws data and 1048 bytes for audio data. The departure data had packet sizes that were distributed in decreasing order with 50% of the data with 8 bytes packet length. The burst length was exponentially distributed, with mean of 2.5 packets per traffic burst.

The high values of the squared coefficient of variation parameter (table 5.1) indicate a very high burstiness of the arrival data. This shows how difficult it is to predict the arrival of traffic data and indicates the observed data's deviation from the Poisson and the compound Poisson processes which would yield a squared coefficient of variation of 1.

5.2 Utilization

The ratio of peak received data to minimum packet length was found to be of the order of 1000:1 for *ws* data which shows a very high burstiness.

Figure 5.1 shows plots of the average transmission intensity against the effective total transmission duration, and of the utilization of the system. The values were calculated for each application program in all the sessions run. Figure 5.1a) indicates a maximum value for average transmission intensity for transmitted data of over 14kbytes/s ($\approx$ 112kbit/s) for combined audio and data traffic. Utilization was measured as the ratio of the total transmission time to the session duration

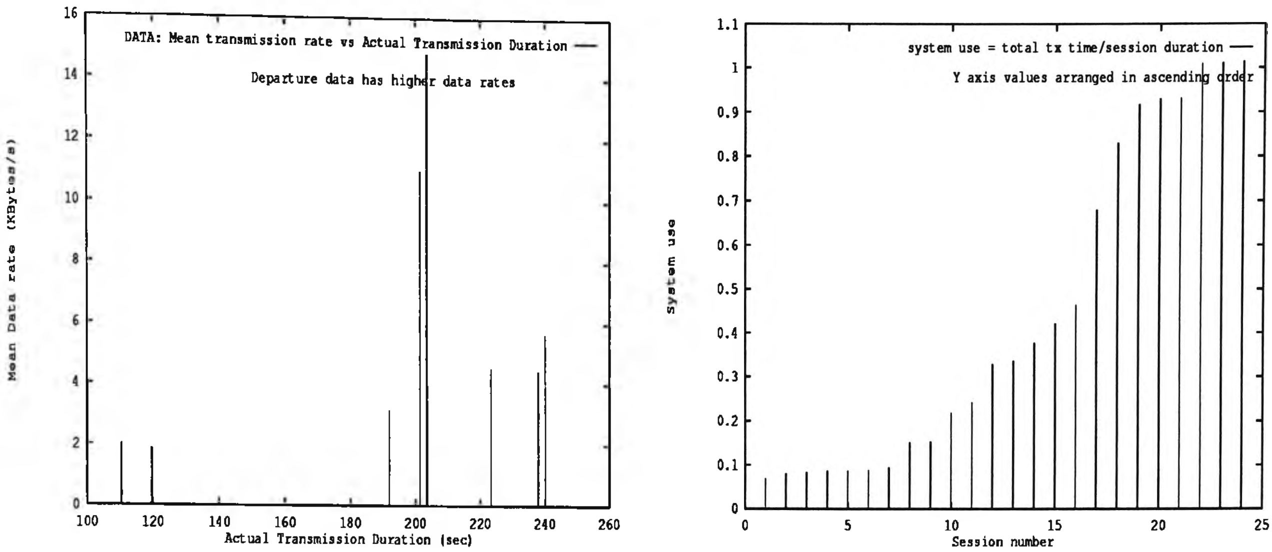

Figure 5.1: *a) A plot of effective transmission time vs channel data rate.*

b) Network Utilization calculated as ratio of effective transmission time to the duration of the session, for each application process

for an application program. From figure 5.1b), the effect of the low average packet length for data ($\approx$ 32 bytes) is to reduce the occupation of the channel to below 50% of the duration of the session. This high ratio shows that the average packet length cannot be used in reserving channel capacity as it would result in unacceptable idle response times when multiple packet transfers are required during a *ws* session. On the other hand, using half the peak rate of 8 kbytes provides a compromise between the maximum packet lengths, (involving the transfer of images, bitmaps), and the under-utilization of network capacity. This results in degradation of conversation quality; it is equivalent to allocating $\approx$ 32 kbits/s when figure 5.1 shows a maximum data rate in excess of 64 kbits/sec.

5.3 Regression Analysis

One way of modelling network traffic is to use the regression analysis to find an equation that relates the required network capacity and other relevant variables to the rate of arrival of the data.

The regression analysis is the process where values of one variable, (e.g. the channel capacity), are predicted using the observed values from some other related variable (e.g. the amount of the transferred data). This is especially desirable when the former variable cannot be directly observed. The analysis, in the end, provides a function

$$y = f(X)$$

where X is the independant variable. This is called the estimated regression equation. The method of least squares can be used to find the regression line fitting the observed data. This method looks for a line where the deviation of data points above and below it is minimized. These vertical deviations are then squared. The equation is useful in planning or forecasting the required processing power for handling a new application, for example. In this research, this analysis could be used to determine how the required channel capacity varied with the number of CSCW sessions running. This assumes that the traffic generated by a user is independant of the number of users in a session, i.e. constant traffic characteristics.

When the traffic generated by one session is independant of that from the other sessions, it is possible to have the traffic characteristics of each session peaking at the same time interval. In that case, the total channel capacity required would be the sum of the channel capacity requirements of each session.

The required channel capacity would be reduced if some dependence exists between the data generated by the running sessions. The fact of whether the sessions' traffic is independant or otherwise is influenced by a lot of parameters. The behaviour of the session users involved affects the resulting traffic generated, as does the network loading, the system software and hardware.

5.4 Summary

It has been demonstrated in this chapter that our characterisation of the traffic model compares well with other models that have been developed so far. Because of the high burstiness exhibited by the arrival process, it was found that the utilization of the network was very low for most sessions. This chapter also introduced the regression analysis as a tool that can be used in further work to find equations that would determine the required channel capacity for a given number of simultaneously-run sessions. The following chapter outlines some of the situations in which our traffic model can be applied.

Chapter 6

Applications

The developed traffic model (Chapter 4 and 5) finds many applications in network design issues like determination of required channel capacity, buffer sizes at network nodes, choice of network protocols and in further investigations using network simulations.

6.1 Simulations

The *opnet* program can be used to simulate the CSCW traffic conditions, as detailed in the previous section, so that a variety of network loadings and several quantities of simultaneously running sessions can be investigated easily and quickly. The information provided in this report provides actual traffic measurements of a CSCW system, and thus can be applied to simulations to predict traffic behaviour in different network conditions and internetwork environments.

To describe the network simulation model it is required to specify[Nutt 82]

- the (bit serial) bandwidth of the transmission medium
- the slot time.

  The duration of the jam period at collisions can be taken as half the slot time.

To describe the load on the simulated network, the number of distinct traffic types (e.g. voice or data) of the connected hosts should be specified. In addition, for each host the following are specified :

- the number of hosts
- the distribution of packet interarrival times with the interarrival time calculated as an interval from the end of one transmission to the beginning of the next
- the distribution of the packet sizes transmitted
- the distribution of the burst interarrival times
- the distribution of the burst sizes transmitted
- fraction of packets which can be expected to contain voice information.

It would be desirable if the network can distinguish between the packets carrying voice and those carrying data. In that case, priotization of voice over data packets could be achieved by using two different backoff algorithms - as suggested in [Nutt 82] where a random algorithm which dynamically determines the backoff time using the uniform distribution can be used for voice sources. This would reduce the delay suffered by the voice communication. Data, which can tolerate larger amounts of delay, would have a binary exponential distribution algorithm applied; an algorithm which degrades the network performance during periods of congestion.

6.2 Channel Capacity

Looking at the results presented in chapter 4, if the allocation of channel bandwidth for CSCW applications on a link between two LANs is based on the measured peak packet length of 8000 bytes, a 64 kbits/s link would be required.

But if the design of the link is based on an average of 4000 bytes, then we need provide only for about 40 kbits/s. The quality of the service would still be acceptable, from our analysis, as the intensity of transmission graphs show that most data is transmitted with data rates lower than 40 kbits/s. Again, this would result in better utilization of the network resources. The penalty for not using peak rate allocation would be unpredictable data losses resulting in erratic pauses in voice conversations. In the end, it is up to the application/system designer to decide if this quality is acceptable.

6.3 Summary

Three applications of the obtained traffic model have been presented. It has been shown how the statistical properties of the various traffic variables can be applied in simulation work. Simulations enable quick and ease investigations using a variety of network constraints. Networks that distinguish between audio and data packets are desirable, in order to use priotization and exploit the fact that data can tolerate delays better than audio traffic. The consequences of using the peak or the average transmission rates in network dimensioning have also been discussed.

Chapter 7

Conclusion

- A traffic model for combined audio and data in a collaborative work system has thus been presented. The model exhibits a random packet interarrival distribution with an exponential burst interarrival distribution. A summary of the standard distribution curves that characterise the model parameters is displayed in table 4.5. The packet size distributions were found to be deterministic for the traffic arrival process, with packet sizes of 32 bytes for *ws* and 1048 bytes for the audio arrival process.

- The model also shows that the transmission intensity for burst segments decreases exponentially with an increase in the amount of data handled. The value of the transmission intensity for traffic bursts was found to be lower than the packet intensity, lower by a factor of 10. This is as expected because the duration of transmission of a traffic burst includes inter-packet time intervals in addition to the effective transmission time for packets. The distribution of the packet intensity indicated normal distributions. This suggested that the application transferred packet data within transmission durations of 15 to 20 milliseconds.

- The arrival process indicates a high burstiness, which makes predictions of the arrival times difficult. The arrival data showed correlation between adjacent arrivals. This appeared to be the reason for the random distribution obtained for inter-packet time interval distributions. The dependence between packets was used to model the arriving packets as batches containing packets whose inter-packet time values were less than a threshold of 42.086 milliseconds.

- The model is different from the traditional Poisson model assumptions, as used in most traffic modelling analyses and teletraffic applications. Poisson arrival processes are independant and memoryless, occuring at random at a mean rate over time. In this research packet arrivals were not independent. The analysis in figures 4.15 to 4.16 show a variable rate of transmission, whilst figure 4.6 indicates that the data arrives in batches. It is the arrival of the traffic bursts that had the statistical properties of a Poisson distribution, with the expected exponential distribution for the inter-burst times.

- The model has been developed using only one of the many CSCW applications. Similar analyses will be needed on different CSCW applications, to test the general applicability of the model.

- Further investigations are needed to check whether similar traffic characteristics are obtained when a different local area network is used, e.g. an ATM local area network, or an FDDI ring. Investigations are also needed for similar checks on networks supplying services which are different from those in a University environment, as was the case in this research.

- Further work could also include exhaustive traffic profile investigations by using the derived model to simulate any desired number of CSCW sessions running on a simulated network whose loading and protocol model can be varied as desired.

Bibliography

[Jander 92] Mary Jander, *Making Voice a Choice for Data networks*, Data Comunications Journal, july 1992

[Tanenbaum 88] Andrew Tanenbaum, *Computer Networks*, Prentice-Hall, Inc, 1988

[Salamone 92] Sal Salamone, *Remote Bridge handles voice and fax calls*, Data Comunications Journal, july 1992

[McQuillan 92] John M. McQuillan, *Multimedia Networking : An Applications Portfolio*, Data Comunications Journal, September 1992

[Radish 92] Radish inc., *WAN/Data over voice - without ISDN*, Data Comunications Journal, June 1992

[Ishii 93] Ishii Hiroshi and Kobayashi Minoru, *ClearBoard : A novel Shared Drawing Medium that supports Gaze awareness in Remote Collaboration*, IEICE transactions on communications, vol. 76, no. 6, June 1993

[Shoch 80] J.F. Shoch and J.A. Hupp, *Measured performance of an Ethernet local network*, Communications of the Association for Computing Machinery(ACM), vol. 23 pp.711-721 Dec. 1980

[Metcalfe 76] R.M. Metcalfe and D.R. Boggs, *Ethernet: Distributed packet switching for local computer networks*, Communications of the ACM, vol. 19, pp. 395-404, July 1976

[Hopper 91] A. Hopper *Design and use of high-speed networks in multimedia Applications* 3^{rd} IFIP Conference on high speed Networking, 1991

[Beadle 92] H.W.P. Beadle, J.Judge, McFarlane and L.M.Hayes *Desktop Teleconferencing for Broadband Networks* MCAT 1992

[Stallings 92] William Stallings *ISDN and Broadband ISDN*, Maxwell MacMillan International, 1992

[Coulson 92] G. Coulson and G.S. Blair and N.Davies and N. Williams *Extensions to ANSA for Multimedia Computing* Computer Networks and ISDN Systems, Vol. 25 No. 1, pp 305-323, 1992.

[Walker 91] John Walker *Netfone : Mike and Speaker* Autodesk SA, Switzerland, July 1991

[Guan 92] Sheng-Uei Guan and Hussein Abdel-Wahab *Multi-user Processes and Shared Capability Lists for Shared Workspace Collaboration* Systems Software, Vol. 19, No.1 ,pp 3-12, 1992

[Lapin 90] Lawrence L. Lapin *Probability and statistics for modern engineering* PWS-KENT Publishing company, 1990, Boston, Massachusets

[Forys 90] Leonard J. Forys and Donald E. Smith *Servicing of Bursty Systems* Computer Networks and ISDN systems, vol. 20 1990, p171-177

[Jordan 90] Larry Jordan *Comunications and Networking for the IBM PC and compartibles* New York, Prentice-Hall Press, c1990

[Nutt 82] Gary J.Nutt and Douglas L. Bayer *Performance of CSMA/CD networks under combined voice and data loads* IEEE Transactions on communication vol. com.30, no. 1, jan. 1982, p6-11.

[Farallon 91] Farallon Computing Inc. *Timbuktu 4.0 User's Guide* , Farallon Computing Inc. Emeryville, CA 1991

[Daigle 86] J.N. Daigle and J.D. Langford *Models for analysis of packet voice communication systems* IEEE Journal on Selected Areas in Communications, vol. SAC-4, No. 6, 1986

[Lauwers 90] Lauwers, J.C; Joseph T.A; Lantz, K.A and Romanov A.L *Replicated Architectures for shared window systems: A Critique* Proc of COIS'90, ACM, New York, pp249-260, 1990.

[beadle 92] H.W.P. Beadle, J.Judge, McFarlane and L.M.Hayes *The Human Side of Multimedia Telecommunications* , MCAT, 1992

[Greenberg 92] Greenberg S,Roseman M, Webster D and Bohnet R *Issues and Experiences Designing and Implementing Two Group Drawing Tools* Proc. of HICSS '92, IEEE Computer Society, Los Alamitos, CA, 1992, pp139-150

[Simon 88] Steve Simon and Trevis Gordon *ISDN Multimedia Desktop Teleconferencing* AT & T Bell Laboratory, 1988

[Ishii 90] Ishii H. *TeamWorkStation Towards a seamless shared workspace* , Proc. of CSCW '90; ACM, NY, 1990 pp13-26

[Habib 92] Ibrahim W. Habib and Tarek Saadawi *Multimedia Traffic Characteristics in Broadband Networks* IEEE Communications Magazine, 1992, Vol. 30, No. 7

[Ishii 91] Ishii and Miyake N, *Towards an Open Shared Workspace : Computer and Video fusion approach of TeamWorkStation* Communications of the ACM, vol 34, No 12, Dec 1991, pp37-50.

[Data 92] Data Comunications Journal, june 1992

[Neuwirth 90] Neuwirth CM, Kaufer DS, Chandhok R and Morris JH *Issues in the design of computer support for co-authoring and commenting* , In Proc. of the Conference on CSCW '90, Oct. 7-10, 1990.

[BT 91] *Computer Supported Co-operative Work* British Telecom Technology, 1991.

[Schwartz 77] Schwartz M, *Computer-Communication network design and analysis* Eaglewood Cliffs, 1977

[Heldman 88] Heldman Robert K, *ISDN in the information marketplace* Blue Ridge Summit, PA, Tab Professional and Reference books, 1988.

[Jackson 70] Fuchs E and Jackson PE *Estimates of Distributions of random variables for certain Computer communications Traffic models* Communication of the ACM, vol 13, December 1970.

[Feher 87] Kamilio Feher *Advanced digital Communications: system and signal processing techniques* Eaglewood Cliffs, N, J. : Prentice-Hall, c1987

[Brand 88] Stewart Brand *The Media Lab : inventing the future at MIT* New York N. Y. USA : Penguin Books, 1988

[Guan 88] Sheng-Uei Guan and Hussein Abdel-Wahab *Shared Workspace for group Collaboration: an experiment using internet and Unix InterProcess Communication* IEEE Communications Magazine 1988 Vol. 26 No. 11

[Johansen 88] Robert Johansen *Groupware : computer support for business teams* New York, Free Press, London, Collier MacMillan, 1988

[Johansen 79] Robert Johansen *Electronic Meetings : technical alternatives and social choices* Reading, Mass. Addison -Wesley 1979

[Rogers 86] Everett Rogers *Communication technology : the new media in society* New York, Free Press, London, Collier MacMillan, c1986

[Wilson 92] Brian Wilson, *Wscrawl* , Hewlett-Packard Co.

[Jain 86] Jain and Routhier, *Packet trains -measurements and a new model* IEEE Journal on selected areas in communications, vol. SAC-4, no. 6, 1986.

[Chen 88] Thomas M. Chen and Davis G.Messerschimtt *Integrated Voice/Data Switching*, IEEE Communications Magazine, 1988, Vol. 26, No. 6

[Fontana 89] B. Fontana and A. Guerrero *Packet Traffic Characterization, Arrival Laws and Waiting Times* Teletraffic Science for new cost-effective systems, Networks and Services, ITC-12, Elsevier Science Publishers B. V. (North Holland) IAC, 1989

[Yang 92] Q.Yang and D.Ghosal and SK Tripathi *Performance Study of two protocols for voice/data integration on ring networks* Computer Networks and ISDN systems, 1992, vol 23, No 1, pp267-285.

[Irven 88] Judith H. Irven and Margaret E. Nielson Thomas H. Judd John F. Patterson and Yoshitaka Shibata *Multimedia Information Services : a laboratory study* IEEE Communications Magazine. 1988. Vol. 26. No. 6.

[Strait 83] Peggy Tang Strait *A First course In Probability and Statistics with Applications* New York, Harcourt Brace Jovanovich, 1983

[Malek 88] Manu Malek *Intergrated Voice and data Communications Overview* IEEE Communications Magazine, 1988, Vol. 26, No.6.

[Sriram 86] Sriram and Whitt, *Characterizing Superposition Arrival Processes in packet multiplexors for voice and data,* IEEE Trans Sept 1986.

[Heffes 86] H. Heffes and D. Lucantoni *A Markov Modulated characterisation of packetzed voice and data traffic and related statistical multiplexor performance* IEEE journal on Selected Areas in Communications, vol. SAC-4 No. 6, 1986.

[Oliver 87] David Oliver translation of *Design and Prospects of the ISDN/ G. Dicenet*, Artech House Boston, 1987

[Stallings 89] William Stallings *ISDN : an Introduction* MacMillan Publishing Company, NY, 1989

Appendix A

Traffic Profiles

Figure A.1 displays a graph showing a typical pattern of the received data from the application programs.

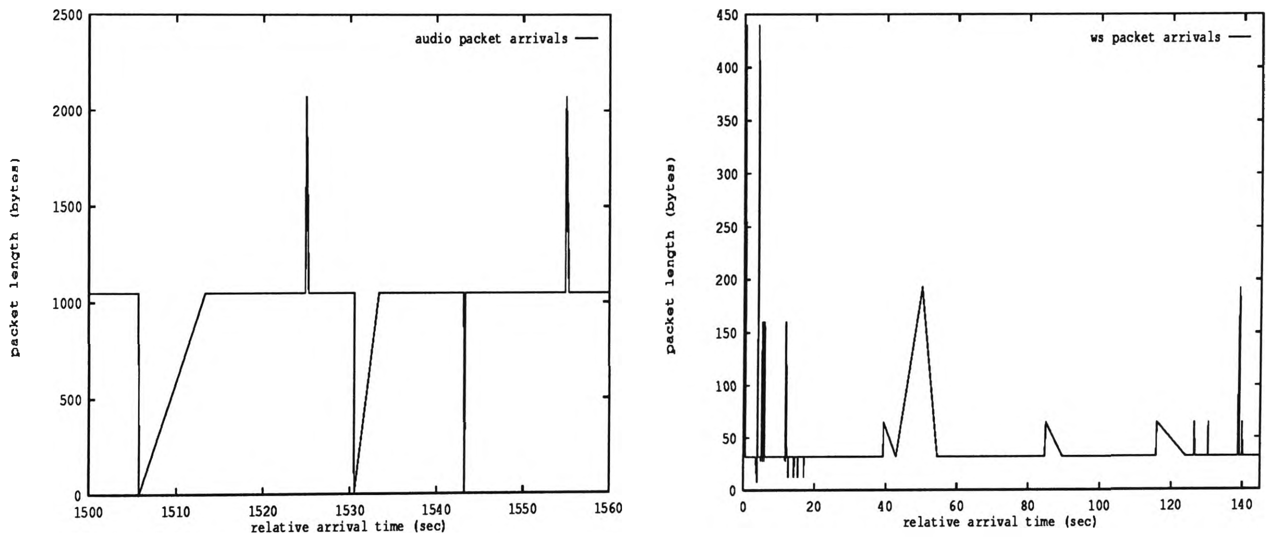

Figure A.1: Typical arrival data: ws and speaker

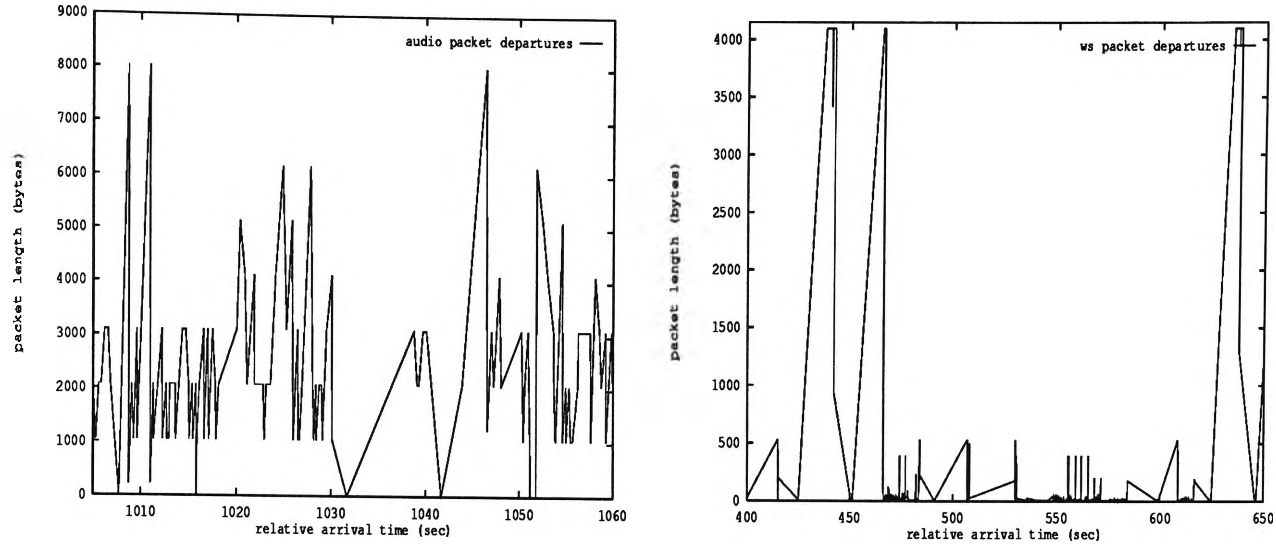

Figure A.2: Typical departure data: ws and speaker

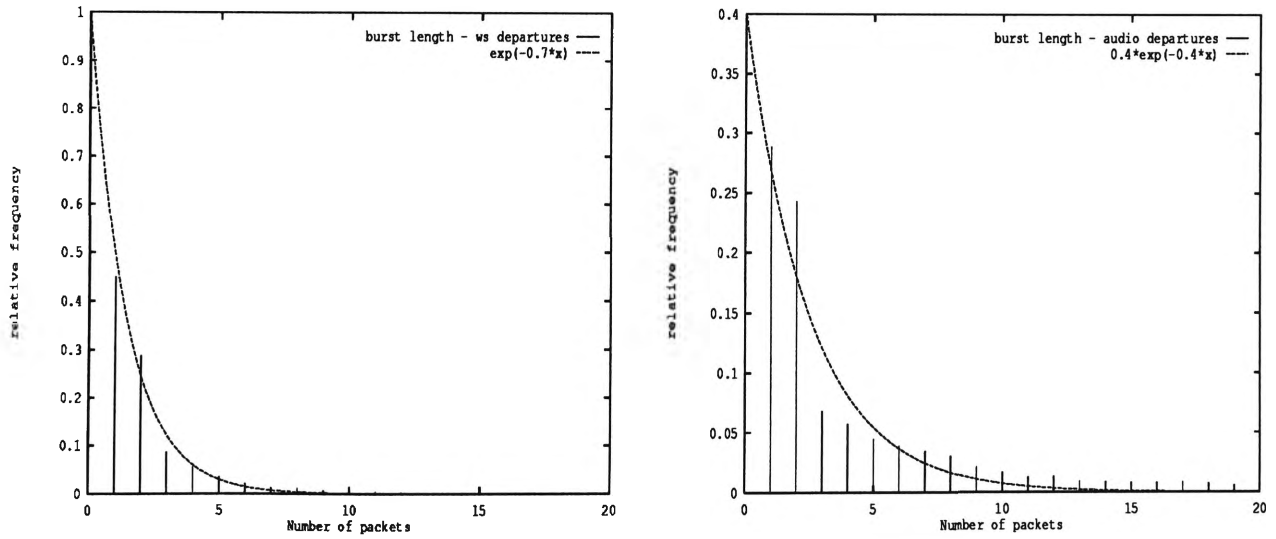

Figure A.3: Burst Length - for departure data from ws and audio applications

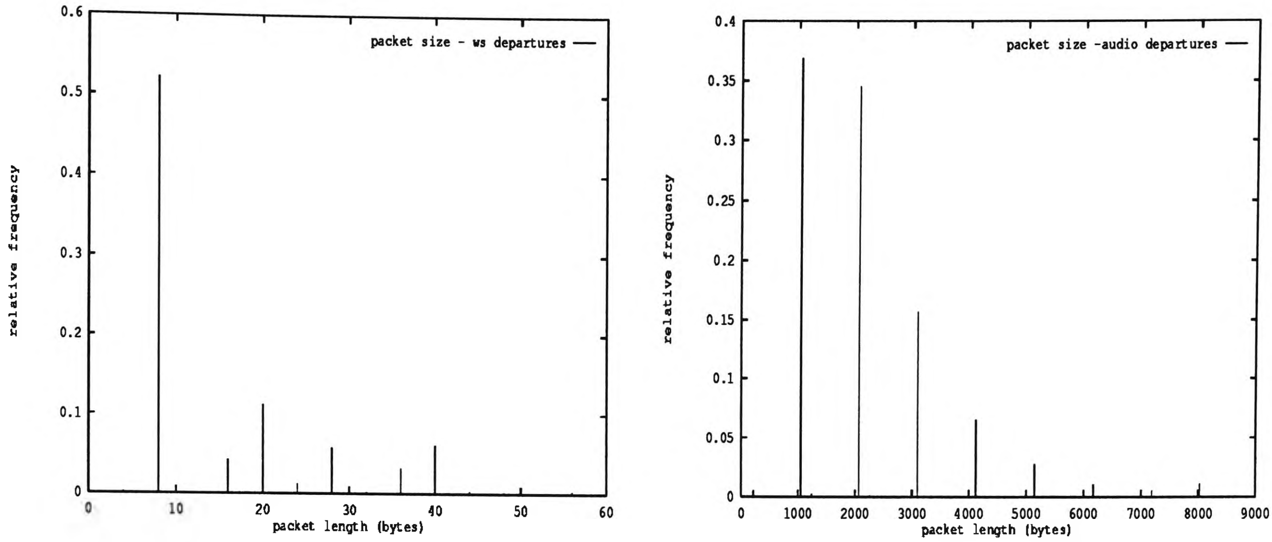

Figure A.4: Packet size Distribution: *departure data from ws and audio applications*

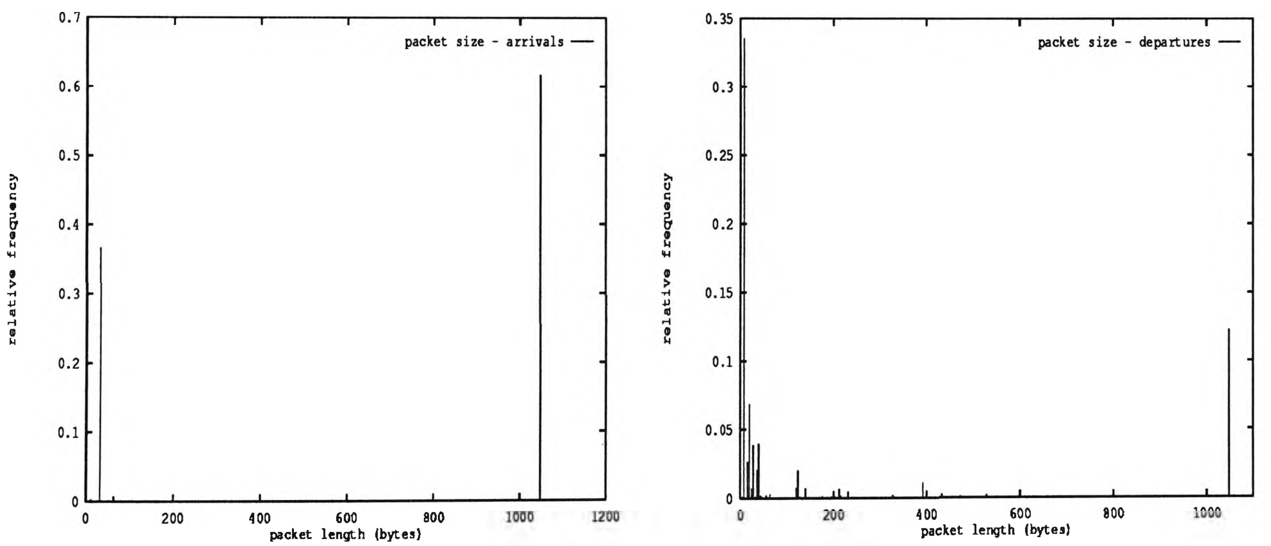

Figure A.5: Packet size Distribution : *combined ws and audio data for both arrival and departure processes*

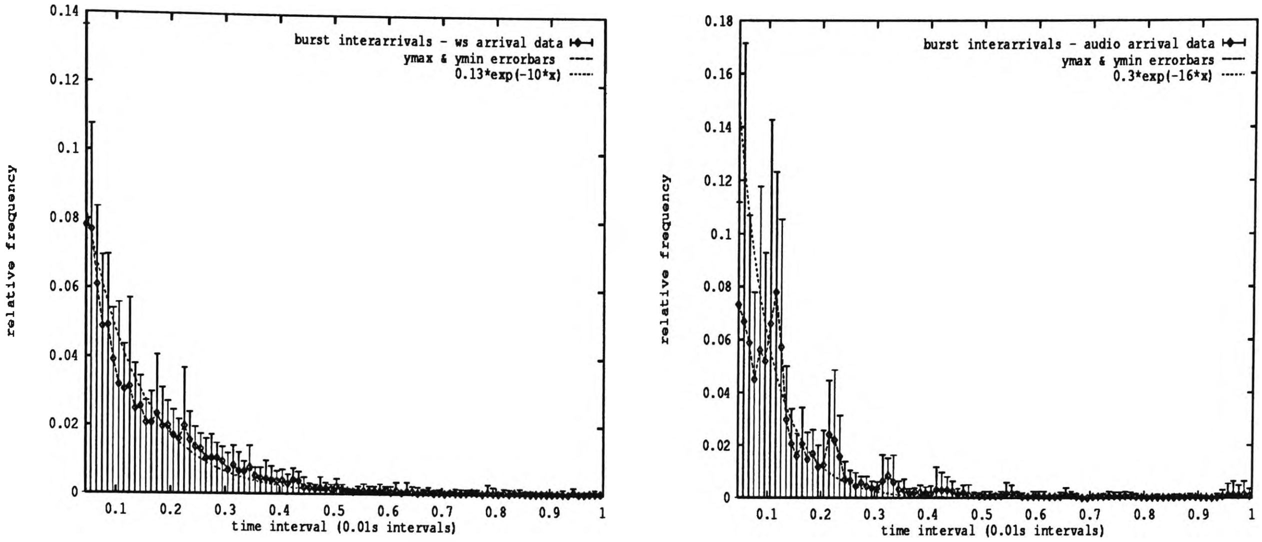

Figure A.6: Burst Interarrival time distribution : ***arrival data for ws and audio applications plotted with the maximum/minimum class frequency used to mark the confidence interval***

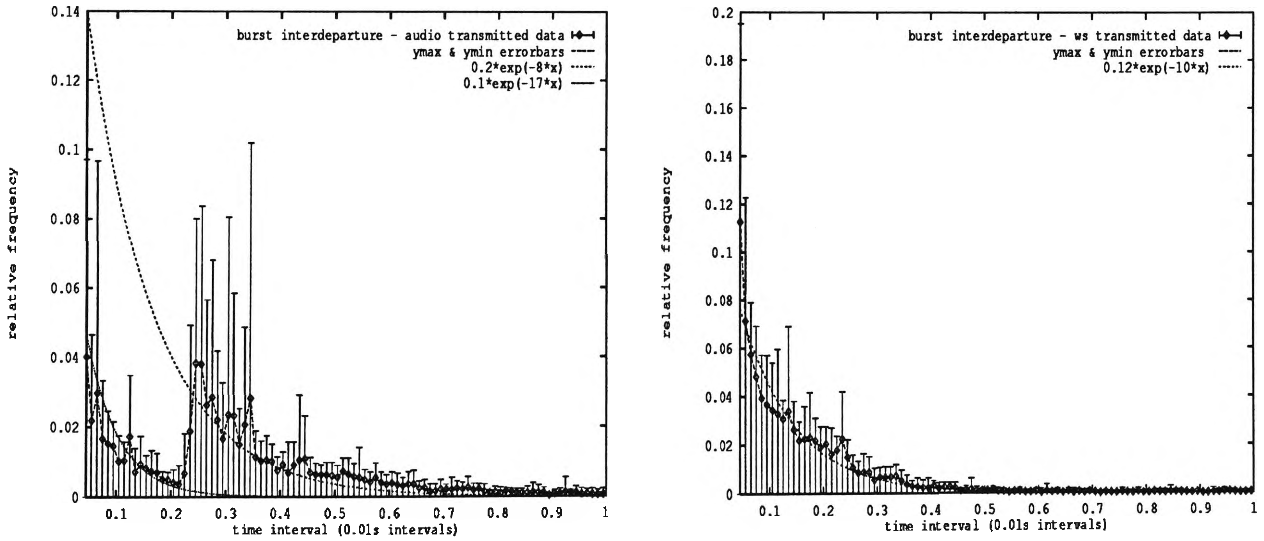

Figure A.7: Burst Interdeparture distribution : ***ws and audio transmitted data plotted with the maximum/minimum class frequency used to mark the confidence interval***

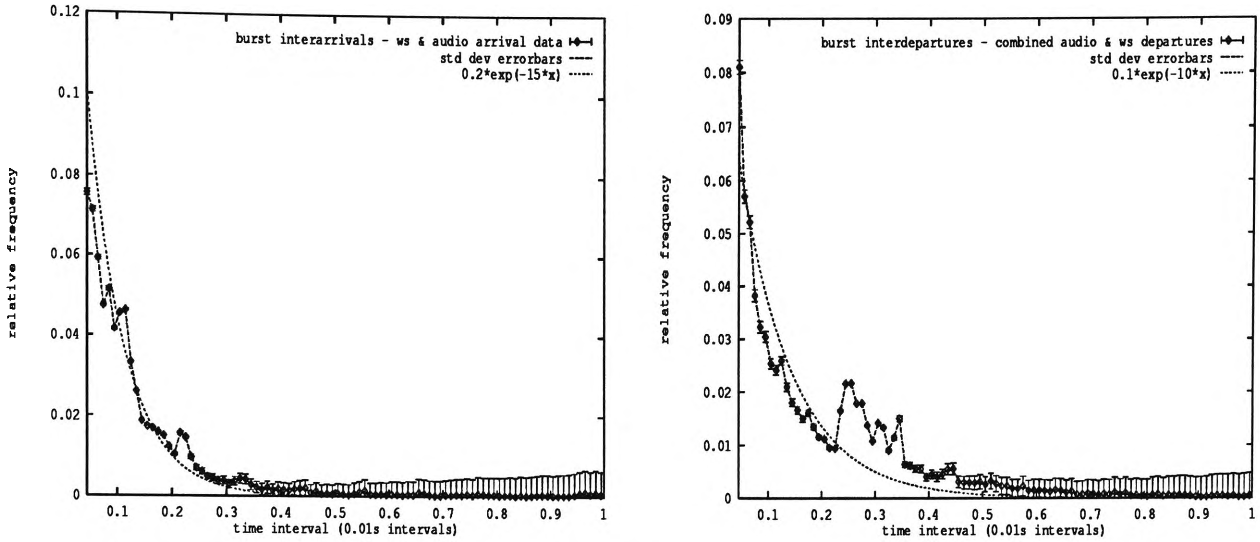

Figure A.8: InterBurst time interval distribution: *for combined audio and data. Both plots shown with the standard deviation as confidence intervals.*

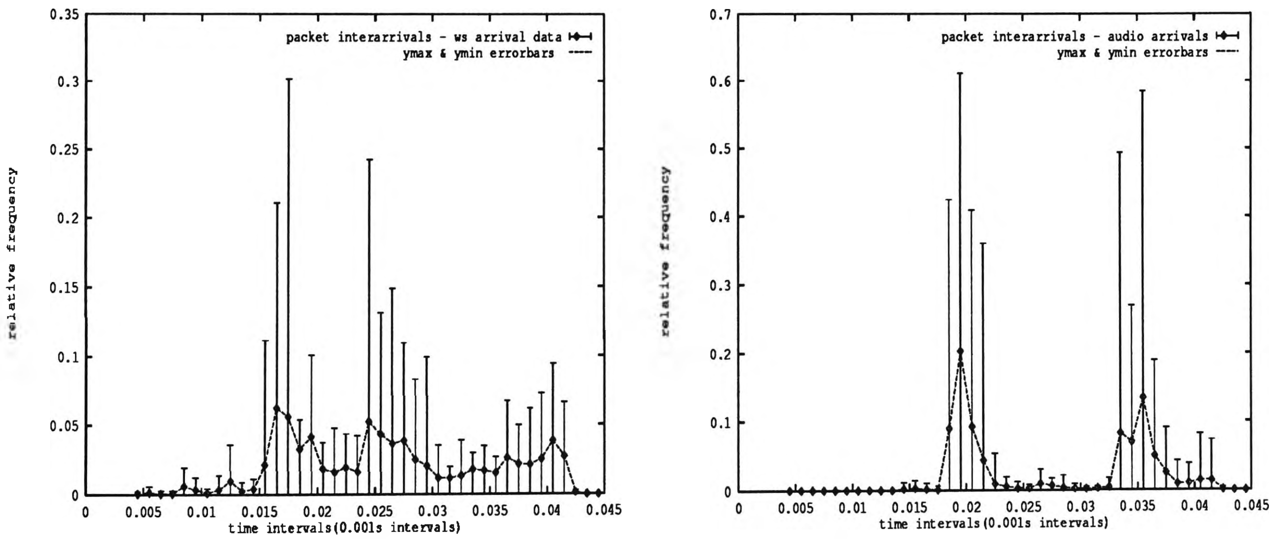

Figure A.9: Packet Interarrival time distribution : *ws and audio transmitted data plotted with the maximum/minimum class frequency used to mark the confidence interval*

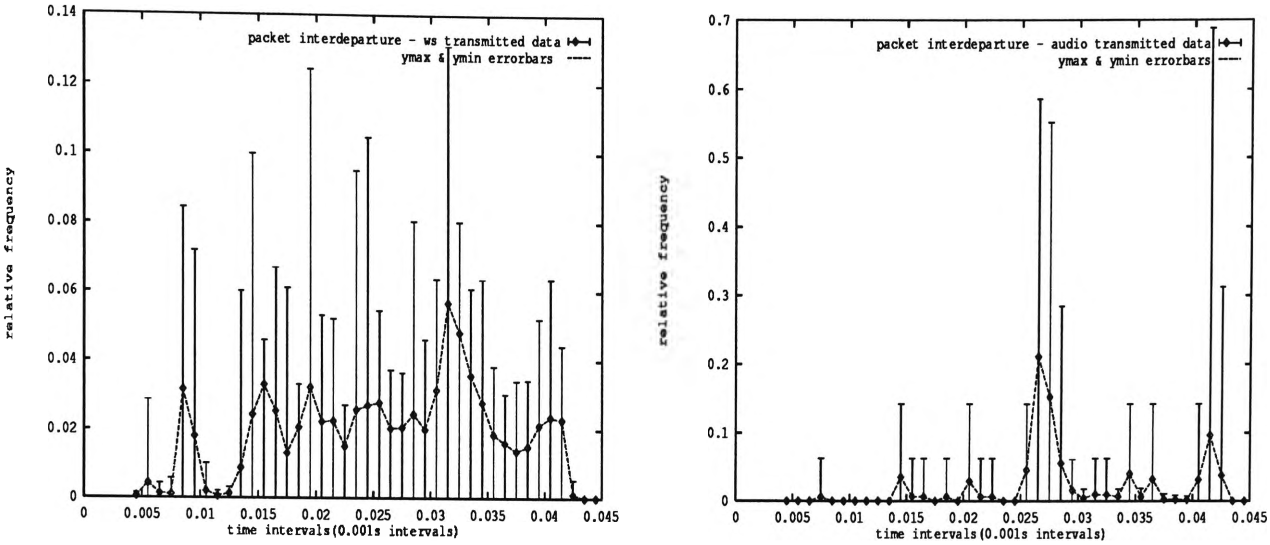

Figure A.10: Packet interdeparture distribution : *ws and audio transmitted data plotted with the maximum/minimum class frequency used to mark the confidence interval*

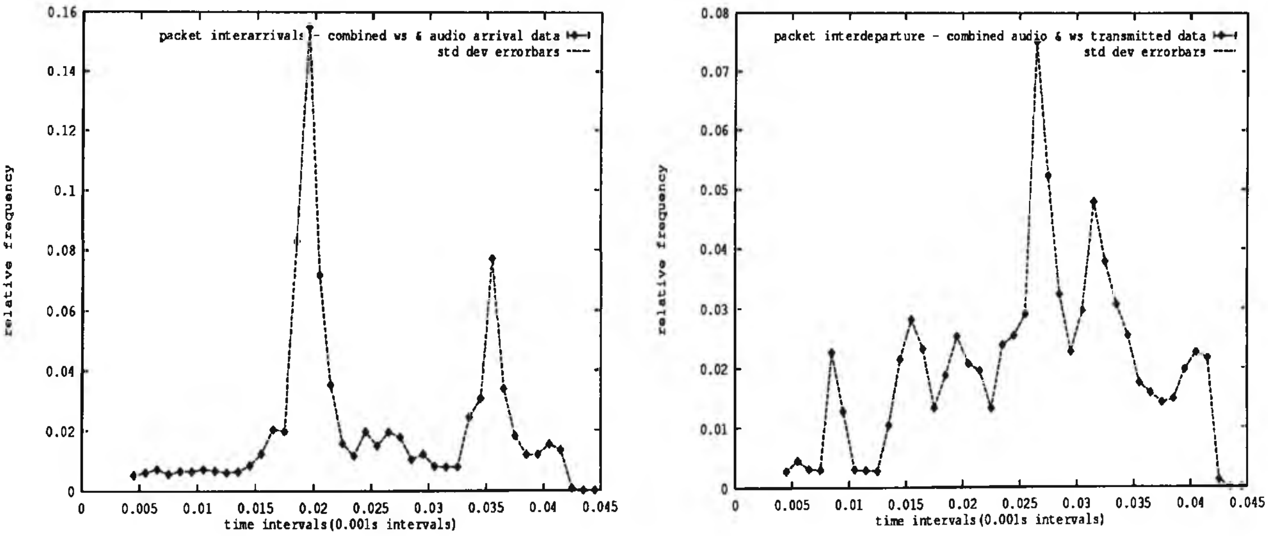

Figure A.11: InterPacket time interval distribution: *for combined audio and data. Both plots shown with the standard deviation as confidence intervals.*

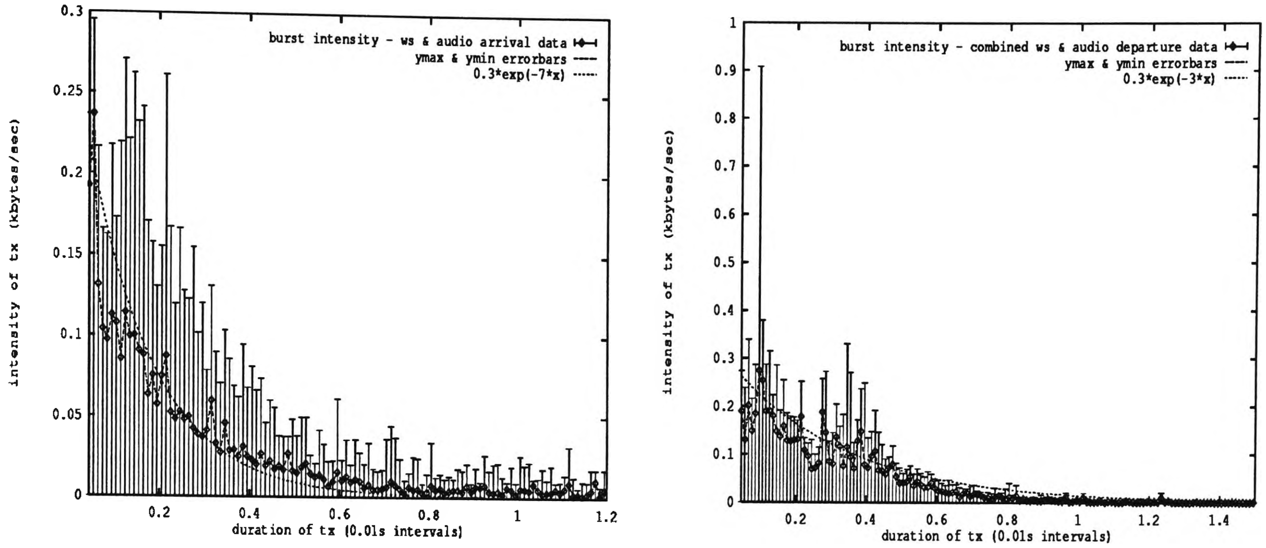

Figure A.12: Burst Transmission Intensity : *combined audio & data with the maximum/minimum class frequency used to mark the confidence intervals.*

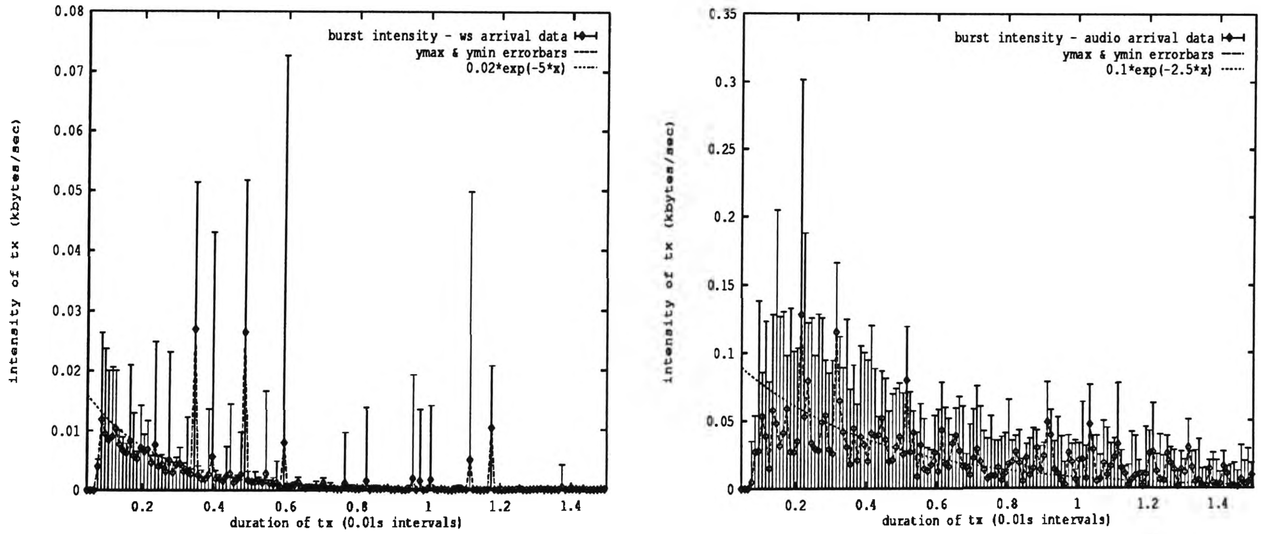

Figure A.13: Burst Transmission Intensity : *for the arrival process of ws and audio data with the maximum/minimum class frequency used to mark the confidence intervals.*

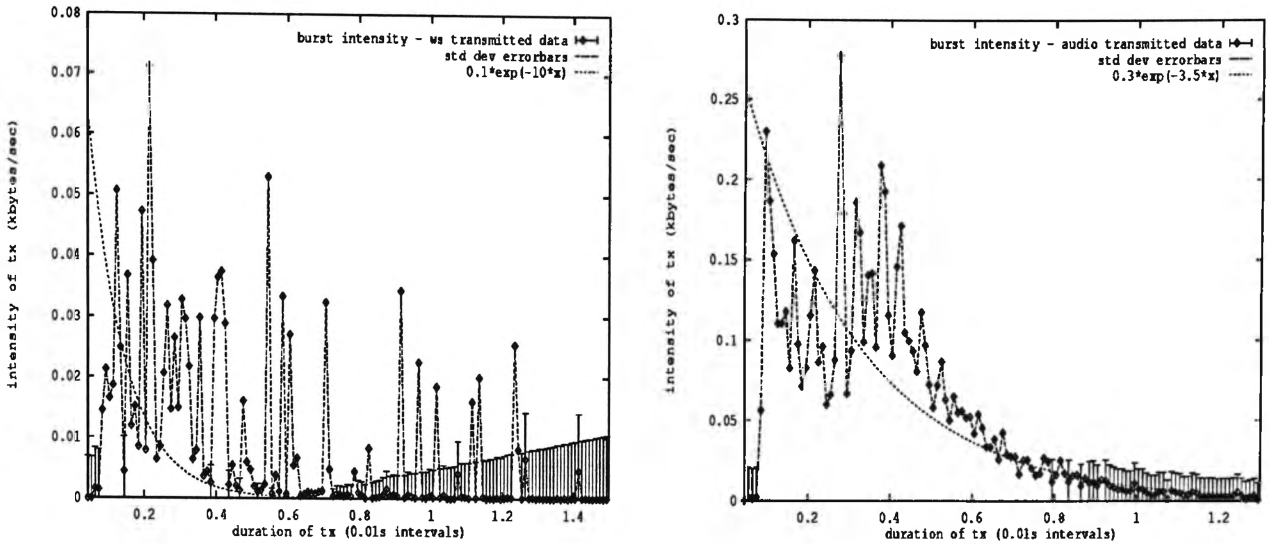

Figure A.14: Burst Transmission Intensity : *for departure data from ws and audio programs with the standard deviation as confidence intervals.*

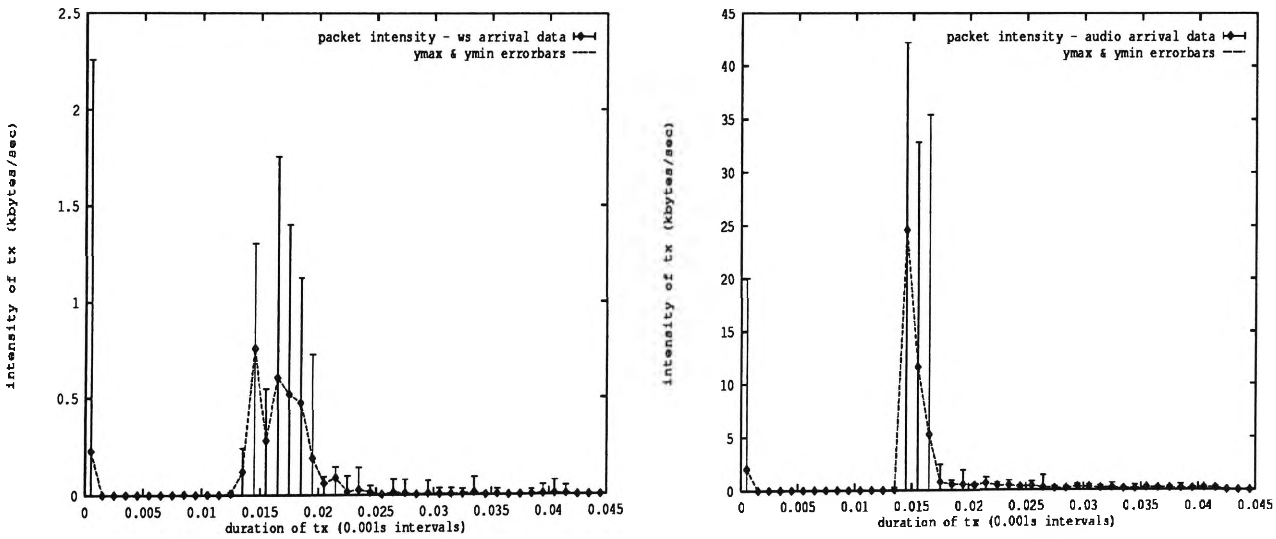

Figure A.15: Packets Transmission Intensity : *arrivals from ws and audio programs shown with the maximum and minimum class fequencies as confidence intervals.*

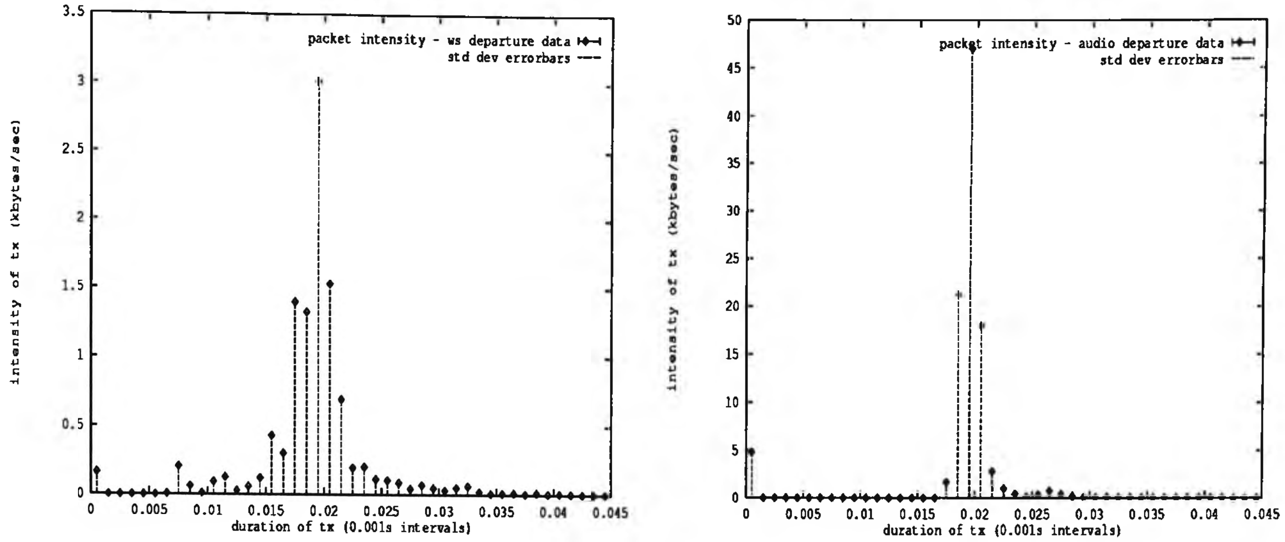

Figure A.16: Packet Transmission Intensity : *for departure data from ws and audio programs with the standard deviation as confidence intervals.*

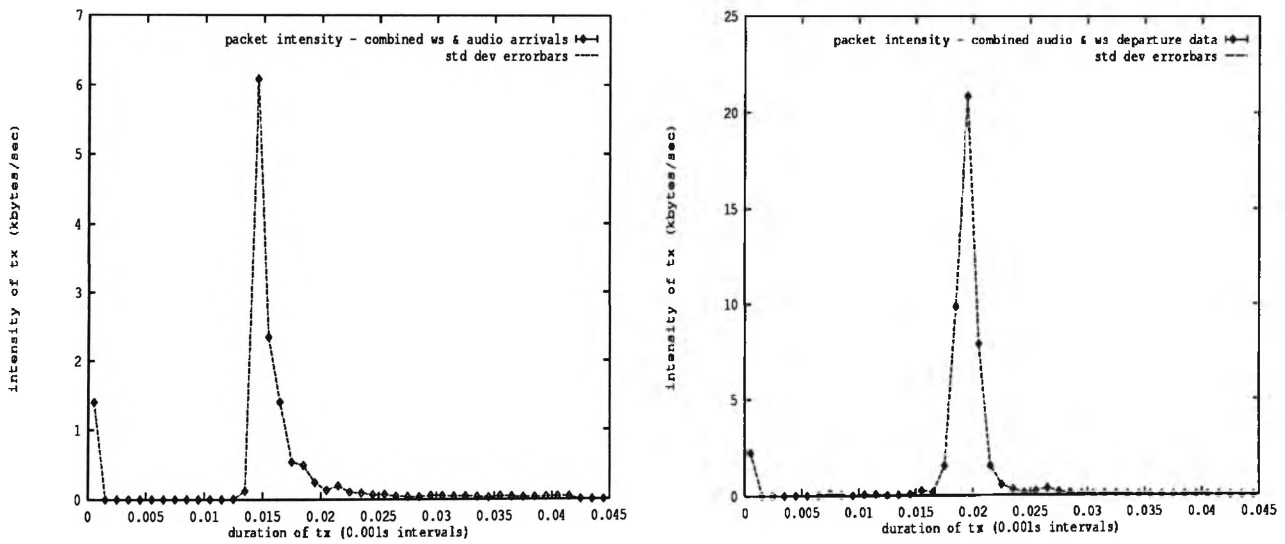

Figure A.17: Packet intensity: *for combined audio and data. Both plots shown with the standard deviation as confidence intervals.*

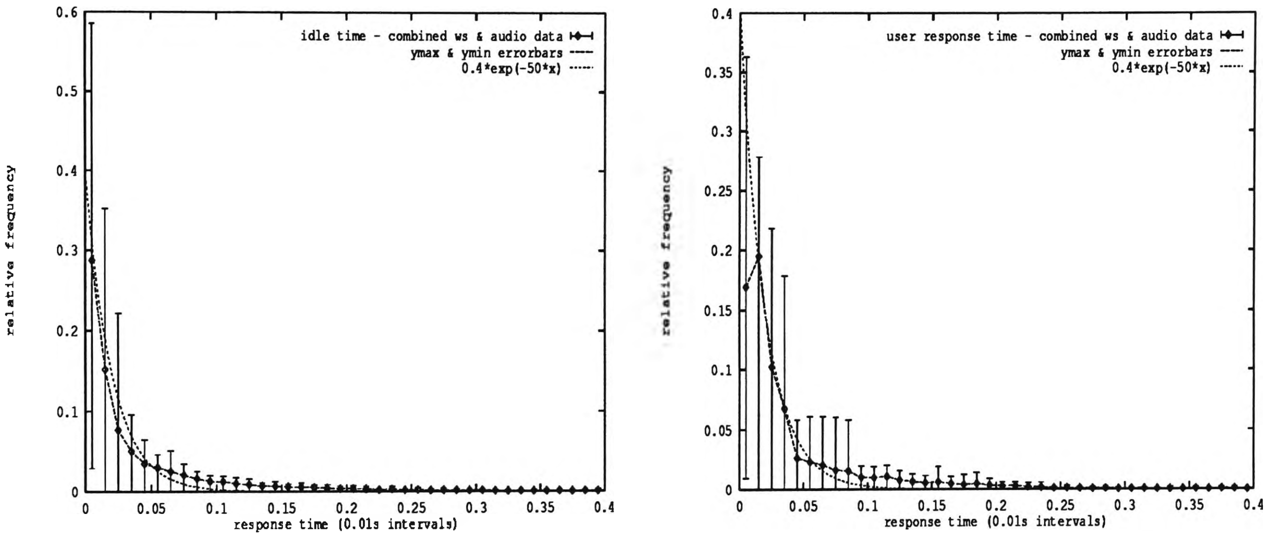

Figure A.18: The user response time and idle time : *for the combined ws and audio data with the maximum/minimum class frequency used to mark the confidence intervals.*

Appendix B

Conferencing systems

Computer conferencing systems, like groupware (section 2.5.2), use the computer as a meeting place for print based exchange of messages among participants located in different places. They are systems in which messages are not sent to another user as an individual, but are sent to a conference which has members.

In off-line computer conferencing systems, upon entering a conference, members are shown all the new material that has been produced since their last visit. This conferencing system is an asynchronous facility in that people drop in when convenient. One disadvantage with this system is it's asynchronous nature which makes it more difficult for interactive communication between conference members. A separate medium e.g. a telephone would have to be used to arrange such a meeting. Example systems are

- on-line computer conferencing systems like *talk* and *write*

- shared drawing surface programs like wscrawl[Wilson 92]

Normally computer conferencing systems are set up between computers with similar parameters. One set is set to originate and the other to answer for one way communication.

A problem with computer conferencing is the handling of interruptions. A participant may signal the end of his/her comments and a need for response, by typing over. The problem arises when more than one participant respond at the same time. This signals a requirement for floor control systems to be incorporated into the conferencing software.

Again most people find it difficult to read whilst they are typing something different. Thus the risk of getting unsynchronized comments is quite significant in computer conferencing systems.

B.1 Audio Conferencing

Audio conferencing systems use an audio channel, e.g. a telephone, to connect the distributed participants. A special conference room, equipped with high quality voice transmission technology is the ideal system sometimes used. The simplest is a telephone set with hands free audio communications and a high sensitivity, which is then used by each group of participants at a site. A similar set-up would therefore be installed at each of the participants site.

One disadvantage with this system is that the participants have to leave their work places for the common teleconferencing room. The other problem in this system is the transmission of visual material from one location to the rest of the participants. This is more costly and often requires the use of special audio-graphic terminals for the transmission of manuscripts. It would be more desirable if a multi-purpose terminal, e.g. the readily available computer terminal, could be used.

Appendix C

Broadband Networks

Most future networks are likely to incorporate Asynchronous Transfer Mode (ATM) networking technology [Stallings 89]. ATM is designed to support throughputs approaching several gigabits per second. It will depend on the setting up of individual point to point links which allow multiple data rates and several cabling types to coexist on the same switch. Because ATM uses fixed length cells[Habib 92], a mix of traffic i.e. voice, data and video can be carried on the same infrastructure. ISDN (described in section C.1) is designed to enable simultaneous delivery of data along with voice calls, making it easy for several users to edit or review the same file e.g. reports, sales figures, etc, at the same time. ISDN's strength lies in the addition of the D channel signalling information that accompanies the call.

C.1 Broadband ISDN

CCITT defines broadband ISDN as a service that requires transmission channels that are capable of supporting rates that are greater than the primary rate[Stallings 89]. The development of BISDN is motivated by the availability of low-cost high data rate transmission media like fibre optics, and the development of cheap high speed modular circuits which can be used as building blocks in the communications equipment. It is also motivated by the emergence of high bandwidth low cost terminal equipment such as computer workstations, video coferencing and video on demand systems.

The requirements for the transmission structure of BISDN are determined by the data rate requirements of the user, and by the services provided. For example, it is required that the network support full motion video as well as narrowband systems. In the design, the duration of the calls is a factor, affecting on the type of switching technology, i.e. circuit switching or packet switching, appropriate for the BISDN service. This call duration parameter can also be used in defining the burstiness of the system as the ratio of the total time during which information is sent and the time for which the channel is occupied. [Stallings 89]

Point to point or point to multipoint service connections can be specified. The network should be able to accommodate different traffic patterns and routing for the same multimedia communications (e.g. voice and data). At the same time it should be transparent for value added services like encryption, speed, and format conversions. It should offer unique signalling channel for each subscriber access point and support multi-rate switched and non-switched connections. Channel bandwidth up to 140 MB/s as in compressed HDTV are designed for, including dynamic allocation of access channels (bandwidth) from the user. The switching facility has to be capable of handling a wide range of different bit rates and traffic parameters (e.g. burstiness). Hence the adoption of asynchronous Transfer Mode (ATM) instead of circuit switching as the basic switching technique for handling the diverse requirements of BISDN. ATM evolves from fast packet switching.

Appendix D

Covariance function

This section gives the definitions of the autocovariance functions used in the above derivation for the IDI. The auto-correlation and the auto-covariance functions are calculated in relation to random variables from the same process.

The autocorrelation function is defined as follows

$$r_x(\tau) = E[x(t)x(t-\tau)] \tag{D.1}$$

The autocovariance for random variables within the same sample is defined as follows

$$c_x(\tau) = \mathrm{cov}[x(t), x(t-\tau)] \tag{D.2}$$

The covariance of any two random variables X and Y is defined as

$$\mathrm{cov}(X, Y) = E[(X-\mu_x)(Y-\mu_y)] \tag{D.3}$$